DOCUMENT RESUME

ED 246 839 IR 011 099

AUTHOR	Akst, Geoffrey, Ed.; And Others
TITLE	Microcomputers and Basic Skills in College: Applications in Reading, Writing, English as a Second Language, and Mathematics.
INSTITUTION	City Univ. of New York, N.Y. Office of Academic Affairs.
PUB DATE	84
NOTE	102p.
PUB TYPE	Collected Works - General (020) -- Information Analyses (070) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC05 Plus Postage.
DESCRIPTORS	Basic Skills; *Colleges; *Computer Assisted Instruction; *Computer Software; English (Second Language); *English Instruction; Higher Education; *Mathematics Instruction; *Microcomputers; Technology Transfer
IDENTIFIERS	Computer Uses in Education; LOGO Programing Language

ABSTRACT

An outgrowth of a conference of developmental faculty and administrators interested in exploring computer-based education, this monograph presents 26 papers devoted to microcomputer applications in college basic skills programs. Included are 5 papers on general topics, 13 on language teaching, 7 on mathematics, and 1 dissent paper. Articles range from broad overviews of the issues and options in this field to research papers and papers focusing on specific software packages. Additional papers explore the use of the word processor for teaching writing, discuss the steps faculty need to take in order to develop their own software or experience in adapting software written by others, and describe curricular models of pairing courses to improve instruction. Two major articles look to the future of the new technology and discuss how the computer may change teaching methods. Specific topics addressed include computerized testing in college basic skills instruction, sentence logic, teaching sentence structure by microcomputer, the potential of computer assisted-instruction in college reading programs, English as a second language (ESL) computer-assisted research and computer-assisted instruction, and relearning mathematics through LOGO. A glossary of technical terms is included. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
- ☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Marie Jean Lederman

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

THE CITY UNIVERSITY OF NEW YORK

BOARD OF TRUSTEES

James P. Murphy, *Chairperson*
Edith B. Everett, *Vice Chairperson*

Paul P. Baard
Blanche Bernstein
Sylvia Bloom
James Cavanagh
Louis Cenci
Armand D'Angelo
Judah Gribetz
Harold M. Jacobs
Jules Kolodny
Melvin E. Lowe *(Chairperson, University Student Senate)*
Robert L. Polk
Joaquin Rivera
Margaret Titone
Henry Wasser *(Chairperson, University Faculty Senate)*
Brenda F. White

Joseph S. Murphy, *Chancellor*

THE COLLEGES

Baruch College—Joel Segall, *President*
Borough of Manhattan Community College—Joshua L. Smith, *President*
Bronx Community College—Roscoe C. Brown, Jr., *President*
Brooklyn College—Robert L. Hess, *President*
City College—Bernard W. Harleston, *President*
City University School of Law at Queens College—Charles R. Halpern, *Dean*
Graduate School and University Center—Harold M. Proshansky, *President*
Hostos Community College—Flora Mancuso Edwards, *President*
Hunter College—Donna E. Shalala, *President*
John Jay College of Criminal Justice—Gerald W. Lynch, *President*
Kingsborough Community College—Leon M. Goldstein, *President*
LaGuardia Community College—Joseph Shenker, *President*
Lehman College—Leonard Lief, *President*
Medgar Evers College—Jay Carrington Chunn, II, *President*
Mount Sinai School of Medicine (affiliated)—James F. Glenn, *President*
New York City Technical College—Ursula C. Schwerin, *President*
Queens College—Saul B. Cohen, *President*
Queensborough Community College—Kurt R. Schmeller, *President*
The College of Staten Island—Edmond E. Volpe, *President*
York College—Milton G. Bassin, *President*

BEST COPY AVAILABLE

ED246839

MICROCOMPUTERS
and Basic Skills in College

Applications in Reading, Writing,
English as a Second Language,
and Mathematics

Spring, 1984

Editor
Geoffrey Akst

Associate Editors
Ruth Davis
Virginia Slaughter

IR011092

The Instructional Resource Center
The City University of New York
535 East 80th Street
New York, N.Y. 10021

© Copyright 1984 by the Office of Academic Affairs
The City University of New York

CONTENTS

Preface — MARIE JEAN LEDERMAN — 1

Introduction — GEOFFREY AKST — 2

GENERAL

Redefining Basic Skills in a Computer Culture — 5
BRUCE VOGELI

Questions Concerning Computer-Assisted Instruction — 8
RITA TERDIMAN and MAX KIRSCH

Micro-CAI for Basic Skills: A New Tool and a New Setting — 13
ELLEN K. LEAHY and ANNETTE PERETZ

Computerized Testing in College Basic Skills Instruction — 17
LINDA J. ALLRED and BERT F. GREEN, JR.

Reflections on Microcomputers — 20
JAN ZANGENBERG and GEOFFREY AKST

LANGUAGE

Word Processing in the Classroom — 22
JOAN BAUM

Word Processing: A Catalyst for the Basic Writing Student — 24
MARIAN ARKIN, BRIAN GALLAGHER, JOAN GREENBAUM, DANIEL LYNCH, KAREN PEARL, and ESTELLE C. SCHNEIDER

Visual Aid for the Writing Teacher: Into the Classroom with a Word Processor — 27
CAROLE McALLISTER

Word Processing in Freshman Composition: One Teacher's Experience — 30
D. TERENCE LANGENDOEN

The Importance of the Teacher in Computer-Assisted Basic Writing Skills Review — 32
WILLIAM R. OATES

SEEN: A Computer Program for Hypothesis Testing in Prewriting — 35
HELEN J. SCHWARTZ

Sentence Logic — 38
MARVIN KUSHNER and ROBERT LATZER

Teaching Sentence Structure by Microcomputer — 43
ROBERT G. GILLINGHAM

The Potential of Computer-Assisted Instruction in College Reading Programs 46
ANNE BRADSTREET GRINOLS

Adapting a Vocabulary Development Program for Use on the Microcomputer 49
REINHART G. KUSSAT

College Reading Programs: Instructors Can Design Their Own 51
ALICE F. RANDALL

ESL Computer-Assisted Research and Individualized CAI 55
GERARD M. DALGISH

An ESL/Computer-Based Project 58
CLARA VELAZQUEZ, ELIE LOWIE, and JUAN VEGA

MATHEMATICS

How Microcomputers Can Be Used to Help Teach Basic Mathematical Skills 60
HENRY AFRICK

Descartes' Delight: A Language for Computer-Inspired Learning in Mathematics 64
JOHN C. MILLER

Basic Mathematics in Colleges and Universities: Computers as a Solution 69
ALFRED BORK

Some Mathematics Software I Have Known--And Hope to Know 72
ELWYN H. DAVIS

Microcomputer Features for Teachers 75
LAWRENCE SHER

Computer-Enhanced Learning in Algebra and Trigonometry: An Eclectic Approach 77
JOAN COMITO, MERYLE KOHN, and ANNA SILVERSTEIN

Relearning Mathematics through LOGO 83
IHOR CHARISCHAK

DISSENT

Earth's Full Color 91
JOHN BELL

GLOSSARY 92

LIST OF CONTRIBUTORS

PREFACE

Picasso is reputed to have said that computers are useless because they only give the answers. Most of what has been written about the use of computers in education, including this monograph, does concentrate on answers, although the questions are implied. Let me state two basic questions. Can computers help college students master basic skills? Can computers help students go beyond simple mastery of skills to those regions in which they become captured by the life of the intellect and imagination?

Microcomputers and Basic Skills in College deals with a number of issues faculty face as they explore ways of teaching through a new medium. For example, what computers can help teach most easily at this moment may not be what is most important for students to learn. Rather than follow the immediate and obvious capabilities of present technology, we must push that technology to its limits through the creation of pedagogically sound software. If computers, unlike the legendary and infamous teaching machines of the past, are to become integral to education, we will need to develop software which is of high quality and not merely an elaborate electronic substitute for workbooks and blackboards. We will need, finally, to do research on the effectiveness of our efforts.

A number of ethical questions are of profound importance. Who will be responsible for the values transmitted through educational software? Who will ensure that the software is not sexist, racist, mindlessly violent? Even more fundamental, will the advent of this brave new world of computers widen or narrow the already existing socioeconomic gap between groups? Will youngsters from poorer homes and schools come to college with experience in computer use even approximating the experience of the more privileged? Once in college, who will have immediate access to computers? Not incidentally, who will develop this software and who will benefit financially and professionally from it? What kinds of partnerships will business and educators form to benefit all--especially our students?

1984 is an appropriate year for educational stocktaking, an appropriate year for discovering how to move large numbers of unprepared college students from the arcade to the academy, an appropriate year to work back and forth from questions to answers to questions.

Marie Jean Lederman
University Dean, Academic Affairs
Director, Instructional Resource Center

INTRODUCTION

This monograph is devoted to microcomputer applications in college basic skills programs. The idea for such a venture emerged from an April 1983 conference, sponsored jointly by The City University of New York (CUNY) and Long Island University, at which hundreds of participants heard talks, viewed demonstrations, fingered keyboards, and exchanged addresses. That conference and this publication enjoy a common goal: to bring together developmental faculty and administrators who are interested in exploring computer-based education.

While about half the articles in this issue are based on presentations which speakers made at our joint conference, the remainder were written in response to invitations extended subsequently to a number of other leaders in the computer-based learning movement. All these leaders--perhaps the term pioneers would be more accurate--seem to share two sentiments: a conviction that even at this early stage of development, computers have a significant role to play in language and mathematics instruction; and a sense of excitement about being personally involved in the creation of the coming order. Scoffers will point out that the history of educational technology bears witness to more than one light that failed, after a brief flicker. However, many of us believe that computers hold the promise to change the way we learn, for good. Because of that promise, it's hard to imagine a more exciting time to be a teacher.

Several of the articles selected for this issue give the reader a broad overview of the issues and options in the field. The paper by Rita Terdiman and Max Kirsch, for example, addresses questions which they, as computer specialists, are most frequently asked by developmental educators. In his paper, Henry Africk explores a wide range of possible approaches to the problem of integrating micros into a basic skills classroom or laboratory, especially in mathematics.

A number of our articles focus on particular software packages which the authors have themselves created. Helen Schwartz, for example, describes a program that encourages students to organize their thoughts during the prewriting process. Marvin Kushner and Robert Latzer explain how their program gives students practice in putting sentences together in order to create units of larger meaning. Robert Gillingham reports on his drill-and-practice software which deals with the exploration of rudimentary sentence patterns. The paper by Jo Ann Comito, Meryle Kohn, and Anna Silverstein describes several arcade-type games which they have developed for use in teaching algebraic and trigonometric concepts.

Geoffrey Akst is Professor of Mathematics at Borough of Manhattan Community College. His recent publications include two Jossey-Bass books, Teaching Basic Skills in College in which he co-authored the chapters on mathematics and program evaluation, and Improving Mathematics Skills for which he served as editor. One of Dr. Akst's special interests is the development of microcomputer programs which facilitate discovery learning.

Lawrence Sher reflects on programing strategies involving joy sticks or paddles that he believes can be of value in the teaching of both mathematics and language. John Miller's paper deals with a programing language of his own creation designed to explore algebraic and geometric relationships through the consideration of graphic displays--a theme related to that developed by Ihor Charischak with respect to the language LOGO.

Alice Randall, writing on a more general plane, analyzes the steps which she believes that faculty need to take in order to develop their own software, and Reinhart Kussat describes his personal experience in adapting software written by others, for use in his basic skills program--themes also developed in a paper by Ellen Leahy and Annette Peretz. Elwyn Davis, a professional reviewer, divulges the names of some of his favorite math software and indicates some directions which he would like the faculty-authoring movement to take. Anne Grinols explores a variety of software concepts for tapping the potential of micros in the teaching of reading.

Two of our articles are research-oriented: the paper by Gerard Dalgish describes a study in which the computer is used to group writing errors in English made by students for whom English is a second language and to relate these error patterns to the students' first language. William Oates reports on an investigation into the effectiveness of using computers in reviewing basic grammar.

Just as the hand-held calculator in the 1970s altered for the foreseeable future the ways in which computation is performed, so the word processor in the 80s is revolutionizing how we and our students write. A number of papers--by Marian Arkin et al., Joan Baum, Terence Langendoen, and Carole McAllister--explore this development from varying points of view.

Several articles deal with microcomputer applications that are not strictly instructional. The one written by Clara Velazquez, Elie Lowie and Juan Vega describes a curricular model in which sections of English as a Second Language (ESL) are paired with sections of an introductory computer course in order to improve student motivation, self-image, and achievement. Linda Allred and Bert Green, Jr. explain how implementing a system of individualized or "adaptive" testing on a microcomputer can make for more efficient student evaluation.

And then there are the seers. Bruce Vogeli peers into his crystal ball and prophesies some ways in which the computer is likely to alter the very skills which we consider to be basic, with significant implications for the curricula of future developmental programs. Alfred Bork foresees and argues for a national effort to develop quality computer-based modules for use in the teaching of remedial mathematics.

Several contributors attack the broad issue of computers from a humanistic perspective: John Bell muses on the importance of keeping one's humanity in a mechanized world, and Jan Zangenberg through his drawings cautions those of us who are crusaders not to take ourselves too seriously.

A glossary of technical computer terms employed in these articles rounds off the volume.

In closing, I would like to express my gratitude to a number of colleagues associated with this publication--and not merely because it is customary. First of all, a word of thanks to the authors, a group of stalwarts who without exception managed to meet our deadlines as well as to suffer our editing. Next, a tip of the hat to Marie Jean Lederman, a closet academic masquerading as a university dean, from the outset lending her considerable talents and support to the project. Goblets lifted to Michael Ribaudo for the sound advice he gave whenever editorial cliffhangers lay in the balance. A salute to Virginia Slaughter whose editorial expertise is well known to readers of Resource. And kudos to Ruth Davis, a new partner and a seasoned editor, who throughout our collaboration showed a rare commitment to excellence. Also at CUNY, it's a pleasure to acknowledge the contributions of Karen Yucht, graphics artist extraordinaire; Garson Wolitzky and company who were responsible for the printing; and everyone in Word Processing who worked on the copy, draft after draft. Lastly, my hand is stretched out to two special colleagues at Long Island University--Lester Wilson and Deborah Rubie--who were major forces in making the 1983 conference on micros a memorable affair.

Geoffrey Akst
February 11, 1984

Bruce Vogeli

REDEFINING BASIC SKILLS IN A COMPUTER CULTURE

It is always a pleasure to meet with colleagues from sister institutions in the metropolitan area. It's a special pleasure to join you at this conference where our task is to examine and study the emerging technology of microcomputers--and to try, through the study of this technology, to predict accurately what the basic skills of the future will be.

Very often, as I am waiting to speak at a conference or a meeting, I am struck by comments made by preceding speakers. In this case, Dean Marotta [Gary Marotta, Dean of Instructional Recource, LIU] commented that microcomputers will "come to the rescue" of basic skills. I think that's an interesting theme, perhaps a theme upon which this conference should be based. It is my belief, however, that microcomputers not only will come to the rescue but they literally will alter the very nature of basic skills.

If through computer technology we are to be able to further democratize education in the culture of the 21st century, then it is clear that the basic skills of computer use, of computer concepts, and of computer thought, must be incorporated in meaningful ways in the educational programs of the present. Otherwise, present students who hope to live and work happily and productively in that future culture will be denied access to the vast quantity of knowledge that will be accessible only to the computer-capable.

It is my theme, then, that we should begin a conference like this one by--if not revoking--at least revising the educational axiom that "experience is the best teacher." Indeed, that axiom has been applied not only in education. It is used in different forms in many areas of human endeavor. Let me rephrase it this way:

> The future will be like the past because in the past the future has always been like the past. Thus, experience in the past has been the best preparation for the future.

That axiom no longer applies in education. I would rephrase it to say that the future will be similar only to a portion of the present. Our job this morning is to define that preimage of the future--that small area of today's comput-

---

Bruce Vogeli, Clifford Brewster Upton Professor, Department of Mathematics and Science Education, Teachers College, Columbia University, is the author of fifty-three books on teaching mathematics and computing. This paper was the keynote address at the April 1983 Microcomputers and Basic Skills conference.

---

er technology that allows us to peer through the veil of the future, to see those skills, those processes, those methods that will be important in education and in the world at large in the 21st century.

As I describe for you some of the kinds of computer-related skills that may replace what we define today as basic skills, you probably will notice that some of these skills are somewhat mathematical. That is no doubt due to professional biases, but it is due also to my belief that the culture of the 21st century will be information-based; and the information environment, insofar as it is a digital environment, will require mathematical competencies in the broadest sense. Requirements will not be limited to arithmetic competencies or algebraic competencies, but will include the mathematical competencies of precise, sequential, and algorithmic thought. What I have done in preparation for my presentation today is to divide the kinds of basic skills that I see as potentially important in the future, into three categories. Although I will discuss these three categories sequentially, I'm not sure that they are hierarchical in the sense that the first category is of a lower priority than the later ones. Indeed, it is perhaps in the first category that we should concentrate our efforts at this time.

The first category is that of technique or operation. Probably most of the basic skills emphasis in today's educational environment is at this level. We are concerned with the mathematical tasks and the communication skills upon which our present society depends. Certainly we will be concerned with techniques and operations in a future computer culture; but I maintain that these techniques and operations may be substantially different from the techniques and operations that we now are attempting to rescue via microcomputer technology.

In the second category, I would like to place those basic skills of the future that I would identify as concepts--things that we will need to understand in order to function effectively in an information-based society. I believe that these concepts are essential before educators can effectively implement educational activities designed to teach the techniques of the first category.

The third and final category is that of knowledge or thought. This category addresses questions of how we know, of why we believe. It is my thesis that these epistemological foundations will be altered in the computer culture of the next century.

Now that I have reviewed briefly the three categories of skills that I believe will be basic in a computer culture, let me describe each category in more detail beginning with the category of technique. Certainly some techniques important today will continue to be basic skills in the computer culture of the next century. Many other skills that we identify as basic now will be somewhat less important however. They will be demoted to the level of valuable, but not essential, skills. Others will, in a sense, be promoted. Of course, I cannot predict accurately all of the next century's basic skills. I shall offer only suggestions of the kind of things that I believe will be important--that may become the basic survival skills of the future.

At the level of technique, there are a cluster of input-output skills that will be quite different from the information transmission skills that exist in today's world. For example, in a computer environment, keyed input--input that is provided by the operator through a keyboard--is a fundamental mode of communication. There are substantial differences between keyed input and the spoken or written transmission of information. Educators will need to direct attention toward training students to utilize keyed input accurately and quickly.

The present mode of information output for computers is primarily visual via cathode-ray display. I believe that this mode will be maintained in the future despite developments in the area of audio input and output. The principal input-output modes in the next century probably will remain keyed input and visual output. There are substantial difficulties associated with assimilating visual output. The rapidity with which the display changes provides a reading environment quite different from that of the printed page. We need to concentrate, as a basic skill, upon the techniques required to assimilate rapidly and accurately the visual outputs associated with computer technology. The input-output skills of keying in and reading out quickly and correctly will be basic skills in any computer-based society.

In a computer-based society, we can expect to be confronted with a variety of codes and coding schemes. In fact, we are inundated with codes already. All of us carry with us in our heads and wallets a multitude of numerical codes: Social Security numbers, student numbers, telephone numbers, account numbers. Today there are few aspects of our lives where codes are not used. Just 100 years ago, the situation was quite different. Our great-grandparents were not burdened with numerical codes other than perhaps their birth dates. For example, during the Civil War, soldiers were not enlisted by serial number. Instead they were listed on military rosters by name and home town.

In the computer culture of the future, information coding will be a way of life. Somehow we must be able to teach students to work with and through numerically coded information. Certainly basic skills in remembering, in processing, in reproducing codes of all kinds, will be fundamental to successful adjustment to a computer environment.

I want to close my discussion of input-output skills with a reference to a skill that is not yet well developed even within the computer community--that is the skill of verification. We will need to find ways of training students to verify input and output. Verification by repetition, verification by visual check, verification by redundant information. In the computer environment of the future, most information that will be communicated by individuals to computers, and by computers to individuals, will contain redundancies in order to provide means of checking accuracy. Although we practice redundancy in speech, that redundancy often is not intentional. It is, in a sense, the result of an urge to talk. Techniques of verification surely will be among the basic skills of any computer-based environment.

(Continued on page 86)

Rita Terdiman and Max Kirsch

QUESTIONS CONCERNING COMPUTER-ASSISTED INSTRUCTION

Computers and their uses in university settings are undergoing a rapid change. The computer-based revolution which is transforming the American economy is also having an increasing impact on education. The rise of computer use in the classroom has produced many questions concerning computer-based training (CBT). CBT consists of two broad areas: computer-assisted instruction (CAI) and computer-managed instruction (CMI). In this article, we will be concerned with CAI, as it is becoming the more popular of the two types at The City University of New York.

What follows are questions most frequently asked of computer specialists concerning the use of computers in the classroom. This list is not exhaustive but rather represents some of the basic issues which are of concern to CUNY faculty working with or contemplating the use of computer-based instruction.

What is available at CUNY for CAI?

Virtually all computer-assisted instruction is provided via interactive computing. Interactive computing came to the universities in the 1960s. Its "user-friendly" characteristics are the primary benefit of this type of computing. That is, the computer communicates with the user by way of meaningful and helpful prompts and responses to his or her input. In addition, the computer responds to commands in a relatively short period of time, allowing the user an environment in which feedback is available to monitor progress and hence increase productivity. Interactive computing is available through microcomputers, minicomputers, and mainframes, which are all available at CUNY for computer-assisted instruction.

These three types of machines differ in the amount of computer power they provide, as well as the nature of instruction they can support. At the present time, the most frequently used mode of computer power is the mainframe, which is a "time-sharing" or multiuser system. A few of the campuses have minicomputers which are used exclusively for CAI, but for the most part within CUNY these machines are older and are in the process of being phased out. Microcomputers, either smaller single-user, self-contained machines, or more powerful multiuser "work stations," are now becoming more popular on the CUNY campuses. They are

---

Rita Terdiman is University Dean for Academic Computing at The City University of New York.

Max Kirsch is Assistant to the University Dean for Academic Computing at The City University of New York.

---

easy to program, exceptionally "user-friendly," and cheaper to use per student hour than mainframes or minicomputers.

Within these computing systems, CAI has been developed via "home grown" methods (i.e., programs written by instructors on either the mainframes, minicomputers, or microcomputers) or through the Interactive Instructional System (IIS) Course Authoring and Course Presentation Facilities available at the University Computer Center or on IBM's microcomputer, the IBM PC.

Various campuses of the University also have purchased commercially available software.

What are the features that I should look for when I evaluate a CBT system?

The features of CBT systems vary; the ones you choose will largely depend on the specific goals you have. Some of the major features include graphics capabilities, color, animation, uppercase and lowercase type, and sound. Other systems have applications which allow students to register on-line, list available courses, generate tests based on the lesson, and synthesize speech. More comprehensive systems also include programs that will produce grades and that chart the progress of individual students.

Who is using and developing courseware at CUNY?

Faculty throughout the University have been employing software for instructional purposes. CAI has been utilized in science and mathematics, as well as the social sciences, languages, and the humanities. The Instructional Resource Center in the University's Central Office of Academic Affairs has been actively tracking the use of CAI in basic skills throughout the University, as well as evaluating the usefulness of certain kinds of software for instructional purposes.

How do I go about obtaining courseware which I can use for my own particular purposes?

There are three ways of obtaining courseware for instructional purposes: you can either purchase commercially available courseware packages, use the course-authoring portion of a CBT system, or you can develop your own. The advantage of the available courseware is that you are required to know very little about computers. The disadvantage is that you will have no control over how the courseware is structured. Many instructors have voiced a desire to develop courseware tailored for their own purposes, even though they are not highly skilled in computer use. One method which has been proven effective is a team approach combining the instructor, or content person, with a computer programmer who can provide the actual writing for the software. The important thing to remember, however, is that it is the content person who plays the dominant role in this team relationship. The most difficult part of software development is in the formulation of the course or lesson; the actual programing is the last step in the process of development. While the individual instructor interested in developing software for CAI should give some thought to the possible limitations

as we know them today, a concentration on the content of the lesson is what needs to be stressed.

What are the major limitations in developing courseware?

The most important limitation to consider is that, at this point, computers are only able to make simple decisions for which they have been programed. Thus, well-written courseware must provide for all contingencies related to possible student responses.

What kind of computer should I use?

The answer depends in large measure on the features desired in the courseware. While powerful mainframes are available at CUNY for CAI through terminals connected by telephone lines, we have found that students often respond more favorably to microcomputers in the classroom. They are easy to use and have highly motivating features. Some machines offer capabilities such as high quality graphics, color, animation, uppercase and lowercase type, as well as music and voice synthesis. Others have huge memories, permitting the manipulation of large amounts of data for generating tests, keeping track of students, as well as other purposes. Finally, they are often, as has been stated, the most cost-effective mode of instructional computing.

If I do use microcomputers, which kind should I get?

A major factor should be the extent to which desirable software is available for each machine. At CUNY, most software development and utilization has been done on Apple computers and IBM PCs, although we are now witnessing a shift towards the purchase of the latter. One reason for this is the compatibility of the IBM PCs with the rest of the University systems, as well as the fact that IBM has quickly captured a very large share of the microcomputer market and is now setting the standards for the industry. However, the pendulum may swing back toward Apple with the introduction of its new computer, the Macintosh.

What kind of additional peripherals should I get with a micro?

This too depends on the kind of courseware desired. Besides monitors, disk drives, and printers, which are standard, some of the other peripherals that are available include videodisks, game paddles ("joy sticks"), touch-sensitive screens, light pens, and speech synthesizers.

Is it important to choose the right computer language when writing software?

The language used when developing software will depend on the expertise of the individuals involved. While some languages may prove to be better adapted for some kinds of orientations than others, there is no fast rule of thumb. What is important, however, is that documentation on the language used, and how the program was put together, exist for the user of the software. Programs should not be written, for example, in obscure languages which will render the material useless once the programmer is no longer available to answer questions or work out the "bugs" that exist.

Much of the software at the University has been written in either BASIC or Pascal, since these are the languages commonly available on microcomputers. For the most part, those without specific computer skills will find authoring systems easier to work with. These systems allow individuals without sophisticated computer knowledge to create courseware that can be used in CAI. The advantage of these systems is that they are relatively easy to use, take less time to develop courseware, and have routines such as answer analysis. The main disadvantage at this point is that each one has limitations in what it can provide for instruction. It is clear that in the near future authoring systems which will allow more flexibility will become available.

Is it true that students are resistant to computer-assisted instruction?

No, quite to the contrary. The experience at CUNY and other universities indicates that students pay more attention to their studies when working with CAI. The reasons for this are varied, but a number of points are important. First, CAI allows a level of individual privacy which cannot be matched in regular classroom instruction. Students can proceed at their own pace, receive immediate feedback, and repeat work in problem areas without the public embarrassment sometimes caused when a mistake is made. Second, the graphics capabilities available on microcomputers have proven to be an effective means to keep students interested in the process of learning. Finally, it is important to remember that today's students have been raised in a society in which television sets and movies are mainstays of American entertainment. CAI is one way in which this orientation can be channeled for educational purposes, and students seem to respond to it well.

How does CAI affect student performance?

CAI is a relatively new method of instruction. It is therefore somewhat early to provide definitive answers to this question. What has been observed, however, is that students using computers do tend to spend more time working on the problems they encounter. Early reports from some of the CUNY colleges (and in particular those using computers for word processing) indicate a great deal of success with the progress of students who are utilizing this method.

Will the computer come to play a dominant role in the classroom?

It is very important to remember that CAI exists only as a tool to supplement a particular curriculum which the instructor has developed. The interaction between student and instructor is, of course, the essential link in the classroom learning environment.

In what direction is the computer industry moving?

Microcomputer technology is progressing very rapidly, with new and higher quality products appearing daily. In general, the near future will witness more courseware available for CAI, as well as more flexible and sophisticated authoring systems. The software as well as the hardware is becoming cheaper and therefore more accessible for classroom use, as well as being more

"user-friendly." Smaller computers are increasingly becoming more powerful, and will therefore have more and varied uses in the classroom.

What are some of the major concerns I should have?

With the proliferation of CAI on college campuses, there are a number of issues which need to be addressed. First, the security of the equipment, usually quite portable, becomes a concern. Second, hardware should not be purchased without first consulting with computer specialists on the particular characteristics and capabilities of various types of machines, including their compatibility with the University's present and future systems. There are companies now which may offer attractive equipment, but which may be out of business next year. It is therefore advisable to consult local computer specialists before embarking on the purchase of hardware. The authors would be pleased to respond to reader inquiries (212-794-5429).

Ellen K. Leahy and Annette Peretz

MICRO-CAI FOR BASIC SKILLS: A NEW TOOL AND A NEW SETTING

Three years ago, ten Apples were installed in the Bronx Community College Learning Center to serve a target group of basic skills students. A basic skills learning specialist and a programmer started investigating commercial products and contacting educators using microcomputers around the country, searching for software that would help adult underprepared students learn. They found very little to meet these students' needs.

At the start, one seasoned instructor offered to furnish the content for a series of drill-and-practice programs. A second teacher came up with an idea for a dictionary skills program. And then a third, months later, asked that his students be tested at the Apple on their weekly vocabulary list. And so it was that teachers became the content specialists--the catalysts. It was the development team, the basic skills learning specialist, and the CAI (computer-assisted instruction) programmer, who made it work.

We found the physical setup of a computer classroom for adult basic skills students had to be different from a lab for science or data processing users. The basic skills lab should be easily monitored, using perimeter installation since students needing assistance may be timid about seeking it out. Using carrels (or other dividers) rather than tables in rows offers users privacy and is particularly important when using the micros for testing (one of the microcomputer's most valuable applications). Viewing other users' screens is not only distracting, it is an impediment to testing-validity. Consoles and monitors should be at heights which prevent arm and neck strain, and adjustable for wearers of bifocals, a condition unique to the adult learner. Printers must be in a stable position with space provided for paper feed and exit to avoid jamming. The printout (the evidence) is of great importance to the basic skills student who needs a study guide when errors are made, or for confidence building when it records a job well done. Temperature control, installation of sound- and static-absorbent materials, elimination of glare, security of equipment--these issues apply to <u>all</u> labs.

Ellen K. Leahy, Learning Specialist, Bronx Community College, has taught elementary school through college and has pursued advanced studies in curriculum development, program evaluation, and applications of computers for instruction.

Annette Peretz, Director, Bronx Community College Library's Learning Center, is a specialist in media librarianship and media applications to education.

Planning for proper staffing of the basic skills computer lab is critical. Patient, knowledgeable supervisors, and assistants with good language skills are the key. Most user problems with the machine are due to his or her use of language in a form unacceptable to the program. In addition, it is only by observing students during pilot testing stages that their responses provide clues to needed revisions, and corrective feedback can be made available when students have problems interpreting CAI programs.

Developing and Adapting CAI Software

Some commercial software, while offering promising possibilities, is only partially acceptable to instructors. For example, the MECC Teacher Utilities Microquest program (1980) inspired a faculty member to input reading-related exercises, but the program was not originally designed to print out student scores. In-house changes on the program brought students hard copies of their completion of ten sets of exercises. A program, Compupoem (1981), developed at another university, was adjusted to print out a copy of a computer-assisted poetry creation exercise which reinforced knowledge of parts of speech. Commercial programs such as Storyboard (1982) and Tic Tac Show (1982) allow instructors to use short corrected in-class exercises which can then be keyed in by students, thus creating relevant drill and practice to accommodate other learning modalities.

It should be noted that most commercial software is copy-protected and unlistable and therefore cannot be adjusted to produce a hard copy of score results. If the program is extremely worthwhile, the lab supervisor should alert the developer to such a need in case a newer edition is contemplated. The developer may even provide a way to customize the program.

At Bronx Community College, faculty play a major role in software development. From a strained meeting three years ago with a single cautious instructor, computer literacy and CAI authoring workshops for basic skills faculty have become accepted and even requested. Sessions on examining and evaluating commercial software enable teachers to identify options in program design. Demonstrations of our standardized formats show them ways of authoring programs more simply.

The BCC experience in basic skills software development has led us to conclude that it is not cost-effective for faculty members to write their own CAI programs unless they have considerable training in programing. We have employed a software design team model consisting of a content specialist, a learning or testing specialist, and a programmer, to produce quality CAI programs in the most efficient way. (See references.)

We have found it best to produce a fairly standardized program to serve as a tutorial, drill and practice, test generator, or actual test with various scoring and output options. In terms of limitations of time and finances, the developing and debugging of a comprehensive program with built-in scoring options, record keeping, and group data management, are justified if other applications

can be adapted to one well-designed model CAI format. We now have three different CAI program formats written for the Apple using the CP/M operating system.

Different Uses of Lab By Basic Skills Students

The lab can be used by individual students for pre- and posttests, vocabulary review tests, practice tests, makeup exams, and drill-and-practice exercises. This kind of utilization results in the greatest student participation. Large group instruction, led by the learning specialist in an orientation workshop, services the largest number of students in computer literacy. This use of the lab can accommodate two or three students at a microcomputer station following a large TV screen demonstration. In addition, an instructor can schedule the lab as a classroom for remedial mathematics, using the computer as a tool, or for writing, using word processing software for revising writing samples. A new use has emerged from a CAI program currently being developed for ESL students. Teams of two students work at a microcomputer using a tutorial demonstration program designed to reinforce rules for spelling words with "-ed" endings. Here the primary purpose is to encourage student dialogue and informal communication about written English. Interesting human dynamics are observed when students learn in this way.

Computer Literacy for Students and Faculty

Prior to assigning students to computer-assisted instruction, arrangements should be made for a large group orientation session to familiarize them with the lab layout, equipment, procedures for using computers, and programs available. At this session students are formally introduced to the major components of the microcomputer, and any special directions for using a specific CAI program. Depending on the class size, students are also given individual or paired hands-on experience inputting their names, course section, student I.D. numbers, and lesson code names. Proofreading input is stressed to reduce confusion about the kinds of messages produced by a computer when a response is spelled incorrectly. The skills represented in these activities comprise a new set of basic skills for our high tech society.

Faculty at BCC are introduced to the micros at an orientation session scheduled for their students, although many attend workshops designed specifically for teachers. The hands-on sessions demonstrate how a "friendly" tool provides out-of-class individual learning, practice, and testing experiences for their students. It also produces personalized records for them.

By now, a growing parade of faculty-authors has begun exploring possibilities for participation in a CAI team project and the microcomputer has become an accepted teaching and learning tool.

REFERENCES

BCC Basic Skills Lab Resource Packet (1983, free). Describes the use of a Learning Center Microcomputer Lab for Basic Skills and includes an annotated listing of Apple software for Language Arts and Mathematics, a software source listing, a periodicals listing, and a software review form. From: Sage Learning Center, Bronx Community College, West 181 Street & University Avenue, Bronx, NY 10453.

Compupoem (1981, $15.95). For Apple II Plus. From: Dr. Stephen Marcus, South Coast Writing Project, University of Southern California, Santa Barbara, CA 93106.

Microquest. MECC Teacher Utilities, Vols. 2, 3, 4 (1980, 3 disks $90*). From: Minnesota Educational Computing Consortium, 2520 Broadway Drive, Minneapolis, MN 55113.

*MECC software is available at reduced rate from N.Y.C. Board of Education, Computer and Information Science Unit, 131 Livingston Street, Room 408, Brooklyn, NY 11201.

Tic Tac Show (1982, $39). From: Computer Advanced Ideas, 1442A Walnut Street, Suite 341, Berkeley, CA 94709.

Storyboard (1982, $35). From: Wida Software, 2 Nicholas Gardens, London W5 5HY, England.

Linda J. Allred and Bert F. Green, Jr.

COMPUTERIZED TESTING IN COLLEGE BASIC SKILLS INSTRUCTION

Computer-administered testing is a practical reality. At a recent convention of the American Psychological Association, many vendors exhibited computer systems that could automatically administer and score the Minnesota Multiphasic Personality Inventory (MMPI) and a panoply of other personality questionnaires and ability tests. In Maryland, the Motor Vehicles Administration routinely gives its written driver's license examination by computer. The Motor Vehicle Administration has found computerized testing a great boon to test security. Formerly, they used four test forms very familiar to the local commercial driving schools. Now each applicant gets a random selection of items from a large pool and can thus be evaluated fairly. The computer can also save time by imposing a time limit. Since the driver's test is strictly a mastery test, the computer can also impose an error limit, stopping testing as soon as the test-taker makes too many errors.

In such examples, the computer administers the same items that would be found on a paper-and-pencil test. Similar testing in the college basic skills classroom is possible. The primary advantages of computer testing are administrative. Presentation and scoring are automatic so the instructor is free to devote more time to individual student needs. In addition, a most attractive administrative advantage is improved test security. Although computers are not larceny-proof, tests are more secure in a computer system than in a desk drawer. With computers, there are no more messy answer sheets, no more ambiguity about students' responses, and no chance that the student will mark the wrong item on the answer sheet. Moreover, the test can be scored immediately and a computer record is automatically available for statistical analyses. And finally, information can be provided by the computer concerning not only total test time, but response time at the item level.

Computers lend themselves to the development of diagnostic achievement testing. Diagnostic testing to determine the types of errors being made by students would allow the instructor to identify the skills areas in which individual students require additional work. Item response times, examination of error pat-

---

Linda J. Allred, Ph.D. candidate in quantitative psychology at The Johns Hopkins University, teaches psychology at Towson State University.

Bert F. Green, Jr., Professor of Psychology, The Johns Hopkins University, is an advisor to an armed services project on computerized adaptive testing, and chairs the American Psychological Association's Committee on Psychological Tests and Assessments.

---

terns, and interactive computerized testing programs that allow the examinee to correct wrong answers could pinpoint individual strengths and weaknesses.

A major development in test theory that can be implemented by computer is adaptive testing. The past decade has seen vigorous developments of the methods of computerized adaptive testing (CAT). This type of testing achieves efficiency by presenting items appropriate to the test-taker's level of performance. No information is gained by presenting very difficult items to a person of limited ability or easy items to very able students, and the experience may be disheartening or annoying to students. Adaptive testing requires a large pool of items that have been carefully scaled for difficulty by pretesting. Further, those items should be discriminating. An item that is of moderate difficulty for persons at average ability levels should be very easy for persons of high ability, and quite difficult for persons of low ability. The pool should have items of moderate difficulty at all levels of ability. This requires a broader spectrum of item difficulties than is customary on a conventional test.

In the typical computerized adaptive test, the examinee is first presented with an item of moderate difficulty. If the person responds correctly, a more difficult item is presented; if the person responds incorrectly, an easier item is presented. After each item, the computer estimates the student's test score based upon information so far, and selects as the next item one that will give the most additional information about the candidate's score.

Both the scoring and item selection are done following procedures based upon item response theory (Lord 1980). The calculations are formidable and a computer is needed. While a person is deciding which answer to select, the computer selects the best next item for each of the two possible outcomes for the current item--a correct answer or an incorrect answer. The computer also calculates the new score for both alternatives. Thus by the time the person has chosen an answer, both the new score and the next item are ready.

For a microprocessor to administer an adaptive test, it must be able to hold in random access memory (RAM) programs for selecting items and scoring responses. Also, the large pool of items must be quickly accessible during the testing. Recent improvements in microprocessor storage capabilities indicate that this may not be a problem on some of the newer devices. An alternative is to use microprocessors as remote terminals for a central minicomputer system where the item pools and programs are located.

One of the most intriguing psychometric advantages of computerized adaptive testing is that examinees can work at their own pace, which yields an ideal power test. Present testing programs have time limits to balance the time that very able students must sit and wait with the time it takes slow students to finish most of the test. In a computerized adaptive test, each examinee stays busy. Each examinee is challenged but not discouraged. Furthermore, when the item pool is large enough, the entire pool can be published without compromising the test. This would make classroom "teaching to the test" less problematic.

Computerized testing is now economically and technically feasible. Equipment costs are coming down, and students are more at ease with computers than

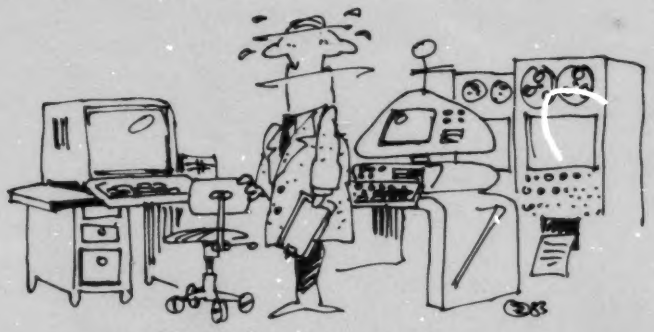
ever before. Although computer-assisted instruction (CAI) can be implemented only by the level of effort needed to write a textbook, giving tests by computer is not much different from preparing paper-and-pencil tests. Adaptive testing is somewhat more complex, requiring special software and many items, but it offers great efficiency for both student and teacher. Educational and psychological testing by computer is no longer novel, and it may soon be commonplace.

REFERENCES

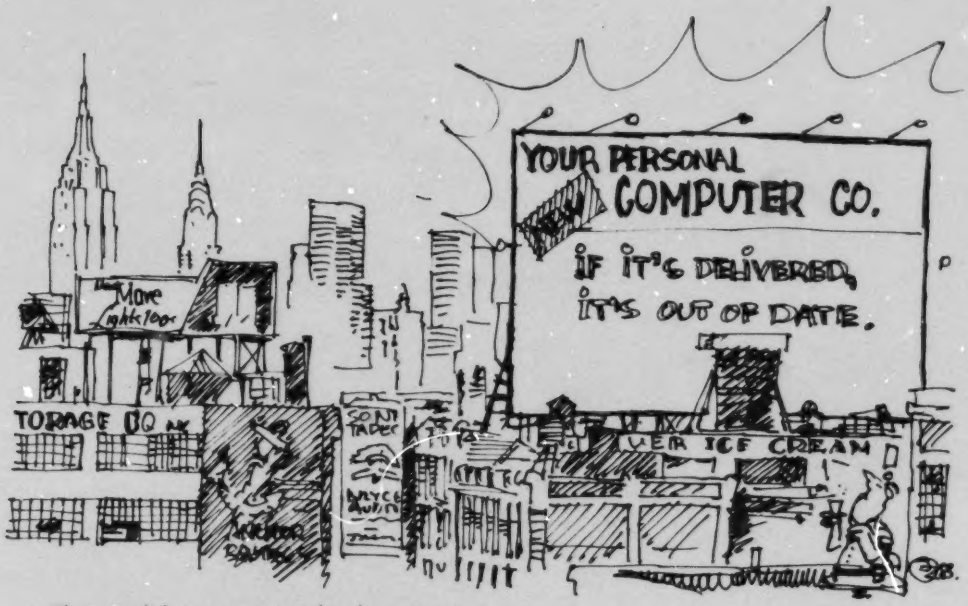
Lord, F.M. Applications of Item Response Theory to Practical Testing Problems. Hillsdale, N.J.: Erlbaum, 1980.

Reflections on Microcomputers

by Jan Zangenberg and Geoffrey Akst



In considering the purchase of a computer system, the overburdened college professor is confronted with too many options.



The models are constantly changing.

Often the services of computer specialists are sought.

Even then, wrong choices are occasionally made.

Jan Zangenberg is one of Denmark's leading commercial artists. His previous work with Geoffrey Akst (editor of this monograph) includes cartoons relating to math instruction which appeared in The Two-Year College Mathematics Journal and the Journal of the CUNY Mathematics Discussion Group.

Joan Baum

WORD PROCESSING IN THE CLASSROOM

Word processing can be used to encourage students to write more, do drafts, and care about editing and proofreading. However, it also provides an important side benefit: students, most importantly those in liberal arts programs, can be introduced to such important job market skills as typing and knowing how to use a computer while they improve their writing and editing skills. In short, word processing can be used as a device to teach both literacy and computer literacy.

It has another main advantage: word processing may be one of the few activities that writing teachers and students can share. Teachers can compare the way they adapt to different word processing programs with the way their students do, and thus discover first hand what works and what doesn't work with different kinds of students. If writing teachers use word processing for their own writing while making it an integral part of a writing course, they will gain immediate evidence about teaching and learning. The usual way teachers come to know about various strategies and procedures, media and methods, is by reading or observing others. Here, however, is a chance for teachers to evaluate their efforts by having them "tested on the pulse"--their own pulse of course, to clarify the Wordsworthian expression--and thus recommend a mode of instruction they use for their own professional advantage. They may take a more critical look at how students learn if they, themselves, run into difficulty with the very same methods or materials.

Trying to get through word processing manuals can be a dramatic illustration of the failure or inadequacy of instruction. English teachers may decide to write their own manuals or introductory notes since many computer programs, including word processing, are written with no particular user in mind or, just the opposite, with every kind of user in mind. If writing teachers get into the act immediately, if only by writing brief explanations about software (or hardware), they will be involving themselves actively with the medium not just plugging it in. Unlike printed texts and anthologies, word processing programs invite creative tampering. Rather than eliminating work for the teacher, and maybe eliminating the teacher, word processing can make the composition teacher's job personal, imaginative, challenging.

Because we think visually--imagining thoughts as pictures in our mind's eye--word processing may be appropriate to help both students and teachers become fluent in thinking and, when the stream of consciousness is muddied, help sift out the valuable deposits.

Joan Baum, Professor of English, York College, is currently working on curriculum development with microcomputers in an introductory journalism course.

Maybe.

Most researchers interested in the relationship between writing and computers would reserve judgment. The field is still too new. Nonetheless, Colette Daiute, of the Computers & Writing Project at Teachers College, Columbia University, notes that evidence so far suggests that students using word processing are less concerned about making mistakes; that they write more, though not necessarily better; and--my own experience to the fore in this regard--that they revise more than they otherwise might when using conventional devices such as a typewriter.

Unfortunately, most research done so far is on younger children. And studies at more advanced levels, such as the effect of Bell Labs' editing program Writers Workbench on the writing of college freshmen and on Bell Labs' scientists and technicians, indicate that greater fluency is not a foregone conclusion. [For more information, see N. H. MacDonald, "UNIX Writers Workbench Software: Rationale and Design." Bell System Technical Journal, 62.6 (July-August 1983), 1891-1908.] Moreover, we're reminded of some disadvantages in using word processing: namely, that it's more expensive than traditional modes; that it's subject to "acts of God" (Con Ed's way of referring to blackouts), and that it's more complicated to use, requiring time on the part of the teacher to provide instruction in typing and in the care of micros and disks.

Then there's the possible danger of confusing looks with substance. Just as it's been shown that some teachers tend to give lower grades to, or entertain lower expectations of, papers that are sloppily presented, so there are dangers in being misled by the neat appearance of copy and by the emphasis word processing puts on mechanical and grammatical editing rather than on logic.

Using word processing to help teach writing will also require instructors to redesign their composition courses. If you're going to change the mode of instruction, then it can't be business as usual. Although the medium is not the message, it will affect the message. It should.

Nonetheless, it seems to me that advantages outweigh disadvantages. With word processing, students can revise as much as they wish, and in private. Of course, if students choose to work in small groups when composing and editing, word processing can also accommodate. It seems an ideal way to address students who have different writing behaviors--some like to do a lot of prewriting, others prefer to edit as they go, and some hate writing altogether but may be seduced by the micro as a gimmick. In fact, some researchers report that it is the weak writer who may benefit most from using word processors; more secure composers, such as English teachers, tend to resist trying a new mode if an old one, such as pen and paper or the typewriter, does fine. In any case, and in my case in particular, there's something delightful about writing at a micro, one's private writing screen; it can remind us of what a personal act composing is.

It is essentially for our students, however, that word processing is particularly relevant. Especially for nonnative speakers and for working adults returning to the world of higher education, writing at a micro can be a good classroom supplement, giving confidence to those unused to writing much of anything and for whom the physical strain and mess of doing revisions are often serious obstacles.

Marian Arkin
Joan Greenbaum
Karen Pearl
Brian Gallagher
Daniel Lynch
Estelle C. Schneider

WORD PROCESSING: A CATALYST FOR THE BASIC WRITING STUDENT

In 1982, out of LaGuardia Community College's student population of approximately 7,000, some 450 students had taken the last course in our developmental program, ENG 099, three or more times without passing. To pass, a student must write a coherent essay of at least 300 words on a previously unannounced topic and make no more than eight major grammatical errors. Testing of twenty of these students showed that sixteen had signs of some form of learning disability. We were looking, therefore, for a means of teaching writing that would enable students to overcome, compensate for, or bypass their disabilities. We were also searching for a procedure that would accommodate two of our most closely held pedagogical beliefs: that writing is a process made up of prewriting, drafting, revising, and editing; and that the key to heightened student interest is collaborative learning, especially peer editing.

We subsequently found that word processing took much of the drudgery out of writing, especially copying revisions. We saw the value of the word processing equipment as its capacity to rearrange words and sentences, facilitate correction, store documents, and format printed output. Students can store successive drafts of written work and retrieve them for editing or printing. A survey of the literature gave further evidence that keyboarding at the typewriter helped some students improve their skills. We felt that word processing might help students break out of a cycle of failure, and the New York Department of Education agreed, offering us Vocational Education funds for the project in 1982-83 and re-funding us for 1983-84.

We planned two sections a quarter, each made up of twelve students who were chronic repeaters. Each student would have a keyboard and screen; each four students would share a printer. At the beginning of the course, professors were available to answer students' questions on the use of the equipment.

The authors are at LaGuardia Community College, CUNY. Marian Arkin, Professor of English, is Director of The Writing Center; Brian Gallagher is Associate Professor of English; Joan Greenbaum, the grant's technical liaison, teaches data processing and economics; Daniel Lynch, the grant's Project Director, is Chairman of the English Department; Karen Pearl is a counselor and learning disabilities specialist; Estelle C. Schneider, Assistant Professor of Communication Skills, does inter-college research in learning disabilities.

Over a ten-week quarter, the course met for two 140-minute periods per week, the first seventy minutes in a regular classroom, the second seventy minutes in the microcomputer lab. A sort of "natural" pattern soon established itself whereby students would complete an essay over three periods. Classroom session #1 would be devoted to generating material, notes, and an outline, with lab session #1 given to doing a first draft. Lab session #2 would produce a second draft, and #3 a final draft. Parts of class sessions #2 and #3 would be devoted to reworking, revising, and polishing drafts, with the rest of these sessions concerned with general instruction and practice in the rudiments of prose writing.

At the outset, we made a decision which proved wise, namely, to teach students the absolute minimum number of commands needed to put essays on the computer. We noted that keyboarding and using a video display screen required a set of eye-hand coordination movements different from handwriting. Almost all students were able, by the third week, to work the machine on their own, needing technical assistance only rarely. Most students began to concentrate closely on the content, phrasing, and organization of their essays, working to improve them from draft to draft. The fact that students had, at the end of each lab period, a neat, very readable printout clearly gave them the sense of a concrete writing product that they could take home and subject to the revising, improving process.

An informal study of student progress over the whole term seems to indicate that working on the microcomputer did not necessarily enable students to avoid misspellings, letter inversions, and omitted words on their first drafts; rather it gave them a system whereby they could and did learn to spot errors and unclear phrasing (first on the screen, then on the printout), and so correct them as they progressed from draft to draft. If anything, the format of this course, which stretched the writing of an essay over the better part of six 70-minute periods, replicated the process whereby "real" writing is done, revised, and redone until the piece is in complete and satisfactory form.

The course's overall curriculum approach was inductive, experiential, and collaborative. For one assignment, students interviewed and wrote essays on each other, checking the veracity of what they wrote from draft to draft with the essay's "subject." For another, students were sent, in pairs, to specific parts of the college (e.g., the faculty cafeteria) to make observations for a descriptive essay. For a third, students watched a film about a neighborhood, then toured their own communities, filling out a questionnaire as they went. Out of this raw material they wrote descriptions of their own neighborhoods.

In all cases, students served as the chief audience for each other's writing. The capabilities of the microcomputer made the clarification of points and the adding of specific details particularly easy once fellow students had identified the exact places such additions were needed. The atmosphere fostered by the use of collaborative learning techniques in the classroom carried over into the lab, with students constantly asking advice and help of each other, as well as of the teacher.

We concluded that the reasons why this course succeeded, far beyond our expectations, were many. Besides slowing students down and getting them to concentrate effectively on writing in sequential drafts, there were personal factors. For one, using the microcomputers was fun and became something of a game (not unlike a video game). More importantly, many of the students--a number of whom had failed the course three to five times--began to experience a sense of accomplishment, first in learning how to operate the microcomputer, and then in seeing how dramatically their writing improved from draft to draft. By the middle of the term, many were writing more than they were required to write, and the essays were significantly more vivid, interesting, and intelligent than the essays written in regular sections of the course. Of the multiple repeaters, 80% passed the course, and most of those passed handsomely, with A's or B's.

In our opinion, there is an important role for the new technology in the writing curriculum. Our experience with students with severe writing problems encourages us to believe that a widespread diffusion of word processing equipment can increase interest in writing among these students.

Carole McAllister

VISUAL AID FOR THE WRITING TEACHER: INTO THE CLASSROOM WITH A WORD PROCESSOR

Until now, the only visual aids available to a writing teacher for displaying either a student's work or writing samples have been the blackboard, and the opaque and overhead projectors. Both these techniques present practical drawbacks. Copying any lengthy material on a blackboard is obviously not feasible so display must be restricted to sentences only. While the overhead can display an entire essay, it does not allow for revision suggestions to be implemented on the spot; editing must be done by hand and at some later time. But now there is a visual aid readily available, a writer's tool which accommodates any size draft and allows for changes or suggestions to be implemented immediately while the text is displayed before the class--namely, the word processor.

The word processor is a computer software program designed to facilitate writing and rewriting because its editing capabilities encourage a writer to revise repeatedly. With a word processor one can swap paragraphs, delete lines, type over errors, and, better still, see those changes displayed in finished form. With these editing features, it is versatile enough to perform any number of open-ended writing exercises as well as capable enough to become one of the most effective pedagogical aids for teaching revision in a classroom situation.

Before the class meets, the instructor types students' drafts onto a disk. Then the instructor displays individual student essays before the class for discussion. Using a revision checklist as a guide, such as the one prescribed by Richard Lanham's Revising Prose (1979), students not only offer suggestions for revision, but actually see the effect of their suggestions in a finished form.

The word processor's capabilities as a teaching aid extend beyond revision. Almost any writing exercise can be easily adapted to the word processor. One exercise that maximizes the word processor's editing skills is paragraph scramble. Simply convert a well-written, coherent paragraph into an unordered list of sentences. The students then try to unscramble the order, piecing together clues from their knowledge of transition. This exercise relies heavily on the word processor's skills in shifting lines, phrases, and typing over errors. The instructor types both the scrambled and unscrambled versions of the paragraph on a disk before class. During class, the students' attempts to form a logical, coherent paragraph are typed from the keyboard onto the disk and displayed be-

---

Carole McAllister, English instructor at Southeastern Louisiana University, developed and now coordinates a microcomputer laboratory for basic skills students. She is working on her doctoral degree at Tulane University.

---

fore them (the instructor or a student can operate the keyboard during class). The students alter the paragraph as often as they need, the word processor typing over errors, deleting a line here, and restoring a line there. The paragraph is shaped and reshaped during the class period, all versions being displayed for continuous debate until the final unscrambling occurs.

Another example of the word processor's effectiveness as a teaching aid is in performing deletion exercises. Here the well-written paragraph or essay appears before the class stripped of some of its essentials--thesis, transitions, conclusions, etc. Working either in groups, individually, or as a unit, the class offers aid to the text. Phrases are added or deleted upon command, with discussion provoked by the paragraph's new look. With editing time no longer an important consideration, the class can offer many new versions to the old paragraph. This editorial assistant fosters group participation in the writing process, for the students immediately see the results of their suggestions displayed in a finished form.

To introduce a word processor as a teaching aid in the classroom, you will need a microcomputer, a disk drive, a monitor, and a word processor. All of this equipment is lightweight and portable and can be moved easily from one room to another on a rolling projector table.

If you plan to buy a monitor, the green phosphorus monitor displays text most clearly and is easiest on the eyes. However, size poses a problem with the green phosphorus since it is presently only available in small models that cannot visually reach a group of students. Depending on the school's resources, there are a number of alternatives available. A forty-five inch television set is a possible solution for those with a spare $1800. But more economically, microcomputers can be connected to virtually any television set with the purchase of a $35 RF/modulator. Some microcomputers will connect to more than one monitor, allowing for the chaining of several monitors together in one class. A computer projector such as that made by Electrohome (about $4600) provides excellent display. However, the most economical solution to the problem is the purchase of a large magnifying screen, such as Beamscope ($70), which will enlarge an ordinary television screen to more than twice its original size. Amplified by this magnifying screen, the size of your monitor's screen is not crucial.

After you have chosen your microcomputer, evaluate word processors compatible with your system. To help you decide which word processor is most appropriate for your needs and budget, there are excellent review articles and books on word processors: Peter McWilliams, "An Introduction to Word Processing," Popular Computing (February 1982) reviews word processors designed for Apple, Atari, CP/M based, and TRS-80 micros. The Word Processing Book (1982) by the same author is a witty, comprehensive review of word processing systems. Creative Computing Software Buyer's Guide (1982) and Consumer Reports (October 1983) review word processors designed for IBM, Apple, Atari, Radio Shack TRS-80, and PET micros.

For teaching writing and revision, you really do not need elaborate editing features; any word processor should be able to delete lines as well as split and

glue lines and type over errors, allowing for speedy revision. Not only should your word processor perform editing tasks well, its editing functions should be easy to learn. Some word processors' editing symbols form a logical pattern in the computer console's keyboard; the names of other word processors' editing functions have high mnemonic value. Further, the word processor should be menu-driven--there should be presented clearly before you a list of functions to choose from: Do you want to set the format of your text? Begin typing and editing? Set your print configuration? Save your text?

The advantages of using a word processor as a teacher's aid are many: First, the search for an appropriate word processor compatible with your own computer system extends only as far as your computer shop. Word processors are available for virtually every size and model computer and in varying degrees of cost and features to accommodate even the most limited budget. Second, a word processor is simple to use, requiring no special computer skills. Following the instruction manual is no more complicated than adding a few more keys to your typewriter. However, as an added bonus, in using the word processor you will gain familiarization with the mechanics of computers with little effort. Further, you need not feel intimidated by a lack of proficiency in programing languages. Adapting exercises to your word processor demands proficiency only in English. Finally, you do not have to rely solely on outside funding to introduce computer learning to your classroom. You need only one microcomputer and one piece of software, not large quantities of each. In addition, if you purchase your own microcomputer for use in the classroom, you can write off your investment as a professional expense. The word processors can become for English teachers one of our most effective teaching aids--visual aids that can enhance our performance in the classroom and encourage our students to write.

REFERENCES

Consumer Reports. 48.10 (October 1983).

Creative Computing Software Buyer's Guide. New York: Creative Computing Magazine, 1982.

Lanham, Richard. Revising Prose. New York: Charles Scribner's Sons, 1979.

McWilliams, Peter A. "An Introduction to Word Processing." Popular Computing (February 1982): 17-31.

------- The Word Processing Book. Los Angeles: Prelude Press, 1982.

D. Terence Langendoen

WORD PROCESSING IN FRESHMAN COMPOSITION: ONE TEACHER'S EXPERIENCE

In Spring 1983, I taught a special section of English Composition I at Brooklyn College. The class was equipped with 24 Apple II Plus microcomputers and 5 Epson MX-70 dot-matrix printers. Each terminal had its shift key modified to work as a normal typewriter shift key, and each was equipped with a 12-inch Zenith ZVM-121 "green" monitor and one Disk II disk drive. The class met from 8:00 to 9:15 a.m. twice weekly but I kept the room open after class from 9:15 to 11:45 to allow students extra practice.

When I first agreed to teach the section, I had only a month's experience with word processing software of a type unusable with equipment at Brooklyn College. So, after consultation with various faculty members experienced in word processing, I chose for my course the software, Word Handler (Silicon Valley Systems, 1625 El Camino Real, Belmont CA 94002), designed for Apple II Plus microcomputers with one disk drive.

Once the master diskette that contains the Word Handler system has been booted, it may be removed from the disk drive and the student's slave diskette inserted. Word Handler permits the display of up to 11 lines and 66 characters on the screen at one time. The lines appear as they do on the printed page. Spacing between lines is indicated by a number which appears at the beginning of each paragraph. A short horizontal bar on the screen between the lines indicates where a page break occurs. Freshman composition students can quickly comprehend the relation between what they see on the screen and what will appear on the printed page. Moreover, the editing and formatting commands are not overwhelming in number and complexity. For all these reasons, this choice of software seemed appropriate for my course. On the other hand, the manufacturer's manual is unsuitable for classroom use; and when the master disk is booted, the screen does not provide information on how to use the system. There are no "menus." Thus I had to write my own set of instructions.

When the semester began, I distributed the first few pages of the training manual I had prepared and immediately got the students working on the microcomputers, entering and editing short paragraphs. At the beginning of each class, I booted the master Word Handler diskette at each terminal, and distributed the students' slave diskettes. After a few weeks, student volunteers were able to help with "start up" procedures.

D. Terence Langendoen, Professor of English and Linguistics at both Brooklyn College and The City University of New York Graduate Center, will have his most recent book, The Vastness of Natural Language, published by Blackwell (Basil) Publisher Ltd., in 1984.

After four class meetings, during which students learned only how to use the word processing system, I assigned a total of six essays (normally about ten essays are assigned in English I classes at Brooklyn College), each due in two-week intervals. The students had to submit a printout of each essay, using any of the classroom's printers and following my prescribed format. Each essay was to be double-spaced and right-justified, with a title centered on a separate line; each page was to include a "header" containing the student's name, the course and section number, the essay number, the due date, and the page number. The preparation of such headers is straightforward using this software. Finally, after I graded each essay, the author would correct and edit the essay and resubmit it within one week of getting it back. Two-thirds of the grade for each essay was based on the initial submission, one-third on the resubmission; and, if no corrections were required (this actually happened in a couple of instances!), the author automatically received "A" or "A+" for the resubmission, as appropriate.

The most difficult problem in teaching the course was providing time for students to use the equipment for assignment preparation. About half the students had classes during the two-and-a-half hours the room was available to them. As a result, I had to allow class time for this use. Because the class was limited to twenty students, I was able to offer each student considerable individual attention concerning matters of style, organization, grammar, spelling, and punctuation. In the Fall, when three sections of this course will be offered, the facility will be available to students for three hours a day, five days a week, and therefore regular class time will be used more effectively for the teaching of composition itself. The three of us who will be teaching the course also plan to develop exercises in grammar, spelling, punctuation, rhetoric, and editing, that we will store on the students' diskettes during the summer before the semester begins.

I had no great difficulty dealing with hardware and software problems during the semester. Apart from a few inexplicable "crashes," the terminals, disk drives, and software all proved reliable. Jamming of paper in the printers was the most common and annoying system-maintenance problem. Throughout the semester, I stressed the importance of "saving" developing essays on diskette. Moreover, short of turning the computer off or removing the slave diskette while an essay is being prepared, one cannot exit the Word Handler system without saving the text file. As a result of the system's design and my precautions, no student lost hours of work because of carelessness or computer failure.

To pass English I at Brooklyn College, a student must not only complete the course but must also pass a departmentally administered and graded exit examination. My students, therefore, had to write a final essay in two hours using pen and ink like everyone else who takes English I. I was gratified that of the thirteen students in my section who took the exit exam, ten passed. Of the three who did not pass, one was extremely erratic in attendance and had not turned in any of the assigned essays. Thus, while the no-show and dropout rate in my class was high compared with other English I sections, the ratio of students who passed the exit exam compared favorably with that of other English sections.

William R. Oates

THE IMPORTANCE OF THE TEACHER IN COMPUTER-ASSISTED BASIC WRITING SKILLS REVIEW

Teacher behavior still may be the most important factor in computer-based learning by students.

In Spring 1979, eight journalism instructors of beginning newswriting classes at Indiana University used computer-assisted instruction (CAI) for a basic grammar review with 162 students. The study's original focus was on the learning achievement of students using CAI. However, differences in the ways that instructors managed the use of CAI in their classes emerged as a significant factor in student learning. The important instructor-differences will be explained after the study environment is described.

The resource for the writing skills review was a CAI curriculum of 30 topics on word usage, punctuation, and grammar. The lessons were on the PLATO educational network at the University of Illinois. (Instructional materials developed for the PLATO system have since become available for the microcomputers of several major manufacturers.) The topics were organized into a curriculum with pretests permitting students to skip those topics on which they showed proficiency. For topics that were studied, the computer administered on-line posttests before students advanced to the next topic's pretest. In addition to the on-line tests, paper-and-pencil tests were given to all students at the beginning and end of the study semester.

In similar studies conducted during a three-year period, average retained improvement shown on end-of-semester tests ranged from 48 percent to 5 percent depending upon the test instrument being used. By 1979, when the number of participating classes and instructors reached eight, data also were collected on various class management strategies employed by the instructors. The CAI review resource was made available to all instructors wishing to use it with their classes, but the instructors (identified subsequently as Instructor A, B, C, E, F, G, H, and K) were free to use it in any way they saw fit.

Three instructors (E, F, and G) made the CAI resource available to students on a voluntary basis. The mean CAI study time for students in these classes was 2.2 hours (SD=1.7, n=63). The mean CAI study time for students in the other five sections which required such study was 3.6 hours (SD=2.2, n=101).

William R. Oates, Associate Professor of Communication, University of Miami, has worked in computer-mediated human communication and computer-based education since 1974. He conducted the study described in this article while at Indiana University's School of Journalism.

Four of the five instructors requiring the CAI basic writing skills review had an explicit deadline for the curriculum of 30 topics to be completed by their students. Instructor H's deadline was 7 weeks into the semester, Instructor A's 9 weeks, Instructor K's 10 weeks, and Instructor C's 16 weeks, which was at the end of the semester. Six of the eight instructors (all but E and G) considered students' CAI work in figuring final grades in the course. Three instructors (F, H, and K) explicitly assigned 5 percent of a student's final grade according to CAI work. However, the bases for grade assignments varied; criteria mentioned were the amount of time spent with CAI, amount of curriculum completed, and scores on the on-line and the in-class tests.

In multiple regression and multiple classification analyses of semester learning, the single factor of the different classroom instructors accounted for 17 percent of the variation in end-of-semester student posttest scores. Also, students' instructors were nearly as important as the students' prior knowledge for predicting these posttest scores.

Fully half of the variation in student achievement accounted for by the instructor factor is explained by the simple matter of whether or not instructors required CAI study of their students. Related instructor behaviors also strongly associated with greater student learning were: setting a deadline for completing the CAI curriculum, and basing part of a student's grade upon the CAI work. Less strongly, but positively associated with student learning were adequacy of computer sign-on information, convenience of scheduling sessions with the computer terminals, and the extent to which instructors checked student progress on the computer and communicated with their students via electronic mail. Instructors credited electronic communication, by the way, for general improvements in student-teacher communication.

Lest CAI be given credit erroneously for the results of traditional classroom teaching on students' performance on end-of-semester tests, the analysis also accounted for the amount of traditional classroom teaching time spent on similar material. Happily, the association between semester improvement and classroom teaching time is positive (Spearman's correlation coefficient=.41) but only half as strong as the association between semester improvement and the requirement to complete the CAI curriculum on a deadline (.83).

While classroom teaching is clearly effective, the more significant point regarding CAI is that it may be effectively substituted for much of a teacher's time in delivering basic writing skills instruction. At no time during the study was the salient question addressed whether CAI or classroom teaching was the more effective method of teaching basic writing skills. In practice, classroom teachers of beginning newswriting have little time to review these basic skills. The important finding is that teachers may use a CAI curriculum such as the one employed in this study as an effective substitute for their own scarce time for tutoring students in these writing skills.

Even with students who are favorably disposed toward CAI as a learning mode, formal motivation seems important for them to receive the maximum benefit. By analogy, the best book on a library's shelves may contribute little to student learning until students are required to read it. More should not be expected of

computer-assisted instruction, although students tend to avail themselves of such instruction quite readily on a voluntary basis. CAI in basic writing skills review appears to be best used when it is integrated into the regular requirements of a course.

Class practices that promote learning, whether related to CAI or traditional teaching, are positively correlated, suggesting that good teachers also tend to make more effective use of CAI resources than do poorer teachers. The association between requiring CAI study and spending classroom time on similar materials (Spearman's correlation coefficient=.42) suggests that both variables may be tapping an underlying commitment on the part of some teachers to help their students master basic language skills through all available means. Other news-writing instructors tended both to give less emphasis in class to basic writing skills and to use the CAI resource less effectively. Some instructors, it would seem, may be more interested than others in helping students overcome their deficiencies. These instructors are also learning that CAI instruction can help.

Helen J. Schwartz

SEEN: A COMPUTER PROGRAM FOR HYPOTHESIS TESTING IN PREWRITING

This essay describes a computer program I wrote, used, and modified to encourage more active learning for students in large English lecture classes. The program asks users to create and test hypotheses and then to refine their ideas by sharing them with others on a computerized network. The program does not produce an essay, but sharpens students' perceptions of data in the field and introduces what the discipline considers to be evidence. This, in turn, constitutes a kind of prewriting for exams or essays. At present, there are two versions of the program available: one for testing hypotheses on characterization in literature, another for hypotheses on analyzing and categorizing a work of art.

SEEN, the name of the program, stands for the Seeing-Eye Elephant Network. The "seeing-eye" tutorial helps a student "perceive" what she or he has "seen"; the student's answers to open-ended questions are recorded (with opportunity for revision), and later "posted" under the student's pen name on a computerized notice-board or network. On the network, the student can see the "notices" of others, comment on those (or his or her own) notices, and read the comments of other student critics. The "elephant" part of the program remembers and records what the student has written and seen, and a printout lets the student (or instructor) "trace" how ideas were developed. The program thus provides a learning and writing environment prescribed in theory: a heuristic for invention (Winterowd 1975; Young, Becker, and Pike 1970) and audience feedback in a nonthreatening environment (Moffett 1968; Emig 1971). The computer can provide these aids better than a workbook can, because computer-assisted instruction (CAI) is interactive and prompting; it is better for some students than a classroom because the computer goes at the student's pace, without the pressure of authority figures or competitors. Moreover, the trace allows student and researcher alike to follow the process of perception: the story of Aha!

Listed below are the format of this analytical program, the wording of the literary application, and selections from a student's actual response (in parentheses):

General format	Particular application (w/ sample response)
Pick an X in a Z.	Name a fictional character X, in a literary work Z. (X = Satan; Z = Paradice Lost)

Helen J. Schwartz, Associate Professor of English, Oakland University (Michigan), has, for the last four years, lectured, consulted, and taught about word processing and computer-assisted instruction in composition.

Create a hypothesis: X = Y	Describe your assessment of character X by completing the following: X is ___Y___. (Satan is tricky.)
Argue that X = Y	Provide evidence to show that X is Y:
--Type A evidence:	What does Satan <u>do</u> that shows Satan is tricky? (He enters the body of a serpent to disguise himself.)
--Type B evidence:	What does Satan <u>say</u> that shows Satan is tricky? (He tells Eve lies about the tree of wisdom to get her to eat from it. . . .)
--Type C evidence:	How do <u>others</u> react to Satan that shows Satan is tricky? (God had originally thrown Satan out of heaven because he didn't trust him.)
--Type D evidence:	How does Satan <u>compare</u> to others in a similar situation and how does this show Satan is tricky? (The other angels do not compare themselves to God and are not thrown out.)*
--Type E evidence:	If there is a 3rd person narrator, what does he say that shows Satan is tricky? (He says that Satan chooses the subtlest of beasts for his embodiment.)
Consider conflicting evidence:	What evidence shows that Satan is <u>not</u> tricky? (Adam and Eve know that there is someone out to get them.)* In light of this evidence, you can now revise your notice, explain the apparent contradiction, or leave the exception without comment. [The student "explained" her contradiction by adding the following: (But they don't know what he will do or what he will look like.)]

*Note that the program does not (and cannot) respond to inaccurate or unclear responses. That is a job for <u>people</u>, the respondents on the network.

After the student finished the tutorial, she saw a notice with a similar thesis (Satan is "clever") on the network and added a comment, signed automatically

with her pen name, Kami. When she signed on for her next session, she found two comments on her notice awaiting her.

The program provides a "skeleton" for introducing conventions in a number of different disciplines. In one test, using the one Apple II Plus microcomputer available to the project, forty volunteers in a world literature class used the program for eight weeks, during half-hours scheduled for their convenience. However, the individual questions can be provided by a teacher who is expert in any discipline. For example, in a second version, a colleague in Art History, Professor Charlotte Stokes, provided the questions for use in an art appreciation class. In this version, the tutorial asks the student to choose a work of art (= X) and an artist (= Z), provide the hypothesis by designating the period or style of art the work illustrates (= Y), and then discuss how the hypothesis is supported by evidence regarding the artist's (a) choice of subject, (b) use of color, (c) arrangement of forms, (d) exploitation of the medium, and (e) treatment of human subjects (if any).

On the basis of field testing, SEEN has been modified. The tutorial closes by prompting students to explain how a character changes throughout the literary work, and to summarize their views. Their work can then be loaded or printed with an Apple-compatible word processing program.

For information about the program's availability, write Helen Schwartz, P.O. Box 911, Rochester, MI 48063.

REFERENCES

Emig, Janet A. The Composing Process of Twelfth Graders. NCTE Research Report #13. Champaign, Ill.: National Council of Teachers of English, 1971.

Moffett, James. Teaching the Universe of Discourse. Boston: Houghton Mifflin Company, 1968.

Winterowd, W. Ross. Contemporary Rhetoric. New York: Harcourt Brace Jovanovich, Inc., 1975.

Young, Richard E., Alton L. Becker, and Kenneth L. Pike. Rhetoric: Discovery and Change. New York: Harcourt, Brace and World, 1970.

Marvin Kushner and Robert Latzer

SENTENCE LOGIC

Since computer jargon is quite common nowadays, you've no doubt heard the phrase "garbage in, garbage out." No great revolutionary statement, but nonetheless true; the quality of the product of a computer (or of any system) is no better than the quality of the original materials.

Sentence Logic, a program designed by the authors for the Apple microcomputer, was conceived of primarily to improve the quality of student writing by providing good quality raw material for students to work with. Since poor quality composition exists at all levels, Sentence Logic runs with content usable by elementary school, secondary school, or college students. The program is designed to create a "halfway house" between computer programs that deal with individual lexical or syntactic units (words, phrases, and sentences) and original writing with a word processor.

Educational programs in English today are common enough. On the one hand, many of them classify words into sentence parts and/or parts of speech. They ask the student to identify individual words but provide little or no context to give the word meaning. Such an approach is analytical; students analyze words and larger units so they can label and classify them. We don't believe this process encourages good writing. Consequently, Sentence Logic emphasizes synthesis: putting sentences together to form units of larger meaning and not belaboring the classification angle.

On the other hand, actual writing by word processing programs, while a boon to the person who handles a pen competently to begin with, may well nonplus the hesitant beginner. Even such "simple" word processing programs as Bank Street Writer, Peachy Writer, and Easy Writer still demand that students put in their own ideas, their own context, their own imagination. For many students, this demand is too great.

The design of Sentence Logic recognizes this problem and provides ideas and a story line for the student to work with.

Marvin Kushner, Professor of Data Processing, Borough of Manhattan Community College, acts as a consultant in his field.

Robert Latzer, Professor of English, Jersey City State College, is currently experimenting with novel ways of teaching writing via the microcomputer.

In standing midway between grammar drills and word processing, Sentence Logic hopes to encourage students eventually to develop their own ideas and their own stories. It hopes to help students "graduate" to a word processor by getting them to put together stories in logical ways using complete sentences and considering alternative ways of phrasing ideas.

This "synthetic" approach to story building has two consequences:

First, it develops in the student a sense rather than just a knowledge of the complete sentence; a feeling which can be depended upon to deal with a variety of complicated grammatical structures (often unclassifiable with traditional grammar) which are frequently encountered in real writing.

Second, it builds a feeling for the continuity of meaning when sentences are joined logically. The idea that meaning is not merely a function of several words put together in one sentence but accumulates from one sentence to the next is important. The sense of that accumulation makes constant rereading necessary and frequent rewriting common.

Sentence Logic develops these consequences in a straighforward manner, simple enough so that the untrained and wary student can become involved in its process. The program's nesting design builds a story of individual paragraphs within which are sentences within which are word groups. Each story begins with a short "preface" introducing the student to the story idea. This initial idea is important because the student can begin to make informed choices the very first time the necessity arises.

After the preface, the program confronts the student with a series of word groups, each containing two complete sentences and three fragments. Each of these 5-group sets is potentially one sentence in a developing story. The sentence is identified by number with the paragraph. Each paragraph is numbered also to show its position in the story.

The student must, in each 5-group set, select one of the two complete sentences. Each is grammatically accurate but one is always more appropriate to the context than the other. If the student selects a fragment, the program tells him what kind of a fragment it is. (This is the only place in which the program gets analytical.) Then it sends the student back to the 5-group set to choose another word group. Eventually the student has to choose one of the complete sentences. At this point, that choice is presented to him with the other complete sentence. As the context of good sentences develops, the student can judge more easily which of the two sentences in any choice is more appropriate to the context. (A "work space" accumulates his choices so that he can see how his choices build the story.)

When the student completes a paragraph of four to five sentences, the entire paragraph is printed on the screen. Since the program concentrates on developing a sense of the complete sentence, and since we want the student to be able to substitute another sentence for any of those he has chosen to build the paragraph, the sentences are presented as individual units rather than in the more conventional linear form. Each sentence has its own line. When the student

sees the entire paragraph, he is given the choice of returning to any of the word groups from which he has chosen a given sentence and replacing that sentence with the other one. He is also given the option of "exiting" from the program or of continuing to the next paragraph.

Choices are crucial here. The student must choose one of the two complete sentences and eschew the fragments. Then a choice must be made, on a stylistic basis, of the more appropriate sentence. As the story line becomes clearer, the student finds the choice of style easier. In essence, this procedure recreates the writing process but supplies raw material for someone put off by having to originate ideas.

Fragments are designed into the program to illustrate seventeen different incomplete sentences. (One of them, the fused sentence, is technically not a fragment; a second one is a logical error.) Most common fragments are the present and past participial phrases and the subordinate clause. Which fragments are used depends on the kind of sentences in each word group. The fragments are classified both by form (as "prepositional phrase") and by function (as "subordinate clause"). Although the classifications may be overlapping and inconsistent, they are easily recognizable by most students who have had any grammar instruction.

The following example of the word-group idea provides the first sentence of the initial paragraph in a story designed for college-level students:

1. Graduating at 17 from high school.

2. When she was 17, Mary Harmon graduated from high school.

3. Mary Harmon graduating at 17 from high school.

4. Mary Harmon, at 17, finished and graduated from high school.

5. At 17, Mary Harmon from high school.

The fragments include two present participial phrases (one beginning with a noun) and a noun plus its modifying prepositional phrase. A student who chooses one of these fragments is provided with its label and returned to the word group set. The student eventually chooses either sentences #2 or #4. Whichever choice is made, the complete sentences in the rest of the paragraph will influence the student's judgment about this one.

In each group, inappropriate sentences show a variety of stylistic weaknesses: passive voice, dangling participle, simple or perfect tenses, wordiness, etc. For example: In one paragraph of a story, the first sentence properly chosen reads, "At lunch, some senior girls were collecting for a wreath." The second sentence gives the student the following choices of complete sentences:

a. The funeral home tomorrow would be sent the wreath.

b. They would send it to the funeral home tomorrow.

Sentence a. illustrates the passive voice. Sentence b. not only shows active voice but also uses two pronouns that have clear antecedents. "They (some senior girls) would send it (the wreath) to the funeral home."

The process makes possible a large number of stylistic subtleties that would ordinarily be ignored in student writing. But because the choices are presented, the student must become aware of their existence. The process might be called "read-writing."

None of the foregoing is to suggest that there is no place for the analytical approach to the teaching of writing. We recognize that analysis is a most important feature of gaining control over one's selection process. Analysis promotes conscious control; without that, no writer can be said to exercise all the options at his disposal.

What we are saying, however, is that at this intermediate stage, when a student wants to create meaning but is wary of having to come up with ideas (in other words, hasn't very much to say), it is more important to emphasize logical relationships among sentences and logical choices of words and phrases than it is to try to create conscious control. It is more important for students to see how meaning grows than to be able to initiate their own meaning.

Indeed, only by developing in students the confidence that they are capable of producing a logically meaningful passage can one be assured that they will eventually try to initiate and develop ideas. Such development is, of course, the primary responsibility of the teacher. Sentence Logic does not squeeze the teacher out of this process. In fact, it encourages participation by both student and teacher. It does so first by providing the student with a printout of whatever he has "composed." This printout should form the basis for a student-teacher conference. Supplementing the printout is a teacher's manual which includes the model passages with which the teacher and student may compare the student's choices. Secondly, the program encourages participation (possibly in a more far-reaching way), because of the nature of the questions implied or proposed by the stories themselves. Here are a few examples:

CARL'S TRAGEDY	How does a high school student deal with a teacher who himself is a hit-and-run driver?
OFFICE PARTY	What kind of morality is forced upon a young lady by business ethics?
SEPARATION	How much influence should a father, experienced in war, have on his son who is about to enlist?

Both because of the raw material used and, hopefully, because of creative teacher influence, when students "graduate" to the word processor, they will be better able to analyze their own creations. But before they get to that stage,

learning to write means learning to build a logical structure and becoming convinced that other people understand what the writer is trying to say.

"Garbage in" does indeed mean "garbage out"; both instructors and students are usually aware of this truism. The instructor's task is not to emphasize so negative an idea but to encourage students, given usable "input," to produce readable "output."

For more information, contact Robert Latzer, 20 Mountainside Drive, Chatham, NJ 07928.

Robert G. Gillingham

TEACHING SENTENCE STRUCTURE BY MICROCOMPUTER

We are beginning to realize the implications of the microcomputer for the English composition class, having already begun to tap its capacities in the authoring process and, even more commonly, for tutorials in the memorative aspects of spelling, grammar, and punctuation. However, not much has been done with syntactical concepts or specific patterns in English sentence structure, perhaps because we tend to see the micro as a rigorously objective machine, while viewing sentence structure as inherently subjective. This brief paper discusses some direct uses of the microcomputer for teaching sentence construction. Simply put, the micro can be an effective tool to help students learn the elementary concepts of sentence structure and to encourage students to apply the syntactical concepts in a formalized act of editing.

I use four types of computer programs (of my own writing) for drill and practice in rudimentary sentence patterns.

Program 1: A Simple Sentence Developer Without Punctuation

This program requires the student to respond to a given part of speech by inserting at a location in the sentence a specific word from that speech group. (My ADJECTIVE shoes are beginning to wear out ADVERB.) Single units of adjectives, adverbs, compound formations, or prepositional phrases are used to develop a fuller sentence.

Program 2: A Bare Sentence Developer With Punctuation

This program allows the student to respond to elements of sentences in terms of separation or enclosure. It forces understanding and practice of such separated sentence constructions as items in a series, coordinate adjectives before a noun, and various introductory modifiers. The user is directed to identify the point(s) of separation and to punctuate the stop. For items that enclose--nonrestricted clauses, appositives, contrasted sentence elements, and following adjectives--the user is to identify the element to be enclosed and then to insert the appropriate punctuation.

Robert G. Gillingham teaches English and computer science at Pennsylvania State University's Fayette Campus. After receiving his doctorate in Old English, his interest in language and communication led to postdoctoral studies in computer science.

Program 3: A Sentence Marker For End Stops

This exercise makes the student identify the parameters of the individual main clauses in multiple sentence environments.

Program 4: A Sentence Developer For Coordination and Subordination

This program, a more sophisticated application of the syntactical concepts encouraged in Program 3, allows the student to apply rules for coordination or subordination (user's choice) by constructing types, delineating parts, and delineating connectives.

These programs, meant to function only as a supplement to actual classroom instruction, enable students to diagnose shapes and to complete components of various standard English syntactical models. Through practice, they learn where to add individual parts of speech to make a bare, unpunctuated sentence more concrete and interesting. They learn to identify the standard and natural boundaries of an English sentence. They learn that elements other than individual parts of speech can be added to a bare sentence, and understand to position these elements (phrases, clauses, etc.) in particular areas. Finally, they learn the shapes that a coordinated or subordinated sentence can take. In short, a consistent use of the above programs can reinforce, in a highly controlled setting, an appreciation and understanding of certain English language sentence models.

In conjunction with the drill-and-practice software, I suggest a regular activity of sentence composition, sentence manipulation, and sentence-to-paragraph editing, in order to let students transfer what they have practiced into what they can, indeed, write. For this purpose, I use Radio Shack's Scripsit package, one of many commonly marketed word processing programs which enable definition and manipulation of written units of various sizes, from letters, phrases, sentences, to multiple paragraphs. For useful evaluations of several word processing packages, see Creative Computing (June 1983).

This segment of my pedagogy is based on composition of freestanding sentences as well as on those in a multiple context, where students can become decisively judgmental with regard to syntactical patterns, their flow, their coherence, and their frequency.

The manipulation of freestanding sentences can, of course, be encouraged in a number of ways and in either a self-paced or a formal classroom environment. I suggest two useful methods: one is to have students write on the cathode ray tube terminal (CRT) a freestanding, simple sentence of their own creation, but instructor-checked. Students are then directed to manipulate those sentences according to the instructor's wishes; e.g., a student might be directed to add one prepositional phrase between the subject and the verb, add one adjective to the sentence, or develop a compound verb and one adverb. A second and equally successful method of treating the freestanding sentence is to have students perform on the micro a series of sentence manipulations with preentered sentences.

For example, I direct the student to call out from the menu and then combine the following into an English sentence of his or her own choosing:

a. I invited Joe home to eat.

b. Joe is my best friend.

c. We ate spaghetti with meat balls.

d. It is one of our favorite meals.

Next, I ask the student to revise his or her combination--repeatedly through the course of a class period--by means of specific syntactical patterns: (1) revise using coordination; (2) revise using an introductory verbal; (3) revise using an introductory clause; (4) revise using an internally placed clause. An excellent text of exercises is Sentence Combining by Clifford and Waterhouse (1983). Discussion of student sentence constructions is appropriate and called for, with the student further noting the effects of positioning on the meaning and emphasis within a sentence.

Finally, and most importantly, I require a daily freewriting and mandatory editing, the latter a less formidable task with a word processor. I inform students that the worth of their papers is determined in part by how sensitively and correctly they have applied a specific sentence pattern with which the class has been working. But I also place much emphasis on their larger obligation in the creative act: to construct word groups that will harmonize meaningfully with the entire writing unit.

In short, I plan that students learn, through drill-and-practice software, the variety of localized sentence patterns at their disposal and apply, by using a word processor, their understanding of the patterns to create an aesthetically pleasing and syntactically mature composition.

REFERENCES

Clifford, John and Robert F. Waterhouse. Sentence Combining: Shaping Ideas for Better Style. Indianapolis: The Bobbs Merrill Co., Inc., 1983.

Creative Computing. 9.6(June 1983).

Anne Bradstreet Grinols

THE POTENTIAL OF COMPUTER-ASSISTED INSTRUCTION IN COLLEGE READING PROGRAMS

Computer-assisted instruction (CAI) has been effectively explored in the areas of science, mathematics, and writing. In the area of reading, studies have been conducted (Fletcher and Atkinson 1972, Mason 1980) which indicate that students taught with CAI demonstrate significant advances in reading skills. Kulik, Bangert and Williams (1983) found that CAI reduced the amount of time necessary for learning. Gerrell and Mason (1983) compared the impact of reading material presented on the computer visual display with the same material in traditional printed passages. The authors concluded that the change in mode had a positive rather than a negative effect. Another positive effect can be the elimination of resistance often demonstrated by nontraditional learners to traditional teaching methods and teachers (Condry and Keith, in press).

Given these and other positive research results, the development of reading software is inevitable. Yet at the college level, available software appears to focus on specific word attack skills (Wedman 1983). As areas of development, multilevel comprehension and critical reading ability have been largely ignored.

The need for such development is obvious. A microcomputer software program, like a controlled reader, can be designed to reveal the text at varying rates of speed, to encourage left-to-right tracking, and to remove previously read material from view in order to eliminate involuntary regression. But it also can do much more. The following is a sampling of the true potential of the microcomputer in the instruction and practice of college-level reading.

Main ideas: Students can be asked to underline the parts of the text which contain the main ideas. The computer can then highlight the main ideas, previously designated by the instructor, so comparisons can be drawn. The percentage of overlap can also be shown. As the students proceed with additional texts, the percentage of overlap should increase, reflecting the student's developing skill in this important area.

Building inferences: Students can be asked questions which require them to draw inferences. If the student's choice is incorrect, the area of the text which contains the information upon which the inference is based is highlighted.

Anne Bradstreet Grinols is Reading Specialist and Coordinator, Reading and Study Skills Program, Cornell University. Her book, Critical Reading and Writing: Reading Across the Curriculum, will be published by Cornell University Press in March, 1984.

The student can then be given another opportunity to make the correct inference, with reinforcement of the proper choice being immediately available.

Critical reading: Students can be asked to identify sections of the text which indicate structural pattern, stylistic devices, or evidence of bias. Student responses can then be compared with highlighted areas of correct response, designed into the program and revealed upon request.

Relationships among ideas: To demonstrate understanding of relationships among ideas, students are asked to construct concept maps, pie charts, or bar graphs, with information from a specific passage. As they begin to understand how ideas fit together, students can better evaluate the significance of what they read. The graphic reinforcement of this usage of CAI adds immeasurably to the learning experience.

Vocabulary in context: Key words in the text are highlighted, and students are asked to demonstrate understanding of their meaning before continuing with the assignment. For example, the student can be presented with a multiple-choice question of definition, analogy, or synonym which he or she must complete correctly for each word. Then the text is returned to the screen to be read and assessed.

Flexibility: Through manipulation of speed, highlighting, or other CAI capability, the student is guided through previewing, skimming, or scanning in such a way that momentum is actually experienced. They can acquire and transfer to their own tests the momentum these skills require once they know what it "feels like."

Analysis: Through comprehension questions with answer choices, students can obtain immediate correction, explanation of why an incorrect answer is incorrect, and another chance to choose the correct one. Immediate reinforcement is the vital ingredient here. With open-ended questions, students are challenged to respond to questions already written into the program, as well as to create their own or respond to those created by their classmates. Students can thus actually begin a "dialogue" in a nonthreatening environment being alone with the microcomputer. From there, classroom written or verbal analysis can evolve using these questions and responses.

One of the most fruitful potentials of the microcomputer would be to enable reading instructors to insert their own material, using texts of their choice, and designating main ideas, inference material, structural patterns, key vocabulary words, etc., into a commercial program designed for such usage. This would eliminate the complaint of the lack of appropriate programs for any student or group of students. Such custom-designed "programs" can be accomplished with minimum expenditure of time and effort. The student's own text can be utilized in a way never before practical.

The advantages of active rather than passive learning, individualization, immediate correction, graphic reinforcement, intensive time on task, and the capability for complex record keeping are as valuable in the teaching of college-level reading as in any other field. Today's challenge is to find and

implement the best uses of current CAI capability, while we search for new uses. The success our students enjoy will motivate them as learners and us as teachers and programmers.

REFERENCES

Condry, J. and D. Keith. "Educational and Recreational Uses of Computer Technology: Computer Instruction and Video Games." Youth and Society, in press.

Fletcher, J.D. and R.C. Atkinson. "Evaluation of the Stanford CAI Program in Initial Reading," Journal of Educational Psychology 63(1972):597-602.

Kulik, J.A., R.L. Bangert, and G.W. Williams. "Effects of Computer-Based Teaching on Secondary School Students," Journal of Educational Psychology 75.1(1983):19-26.

Gerrell, H.R. and G.E. Mason. "Computer-Chunked and Traditional Text," Reading World 22.3(1983):241-246.

Mason, G.E. "Computerized Reading Instruction: A Review," Educational Technology 20.10(1980):18-22.

Wedman, J. "Software: What's in It for Reading?," Journal of Reading 26.7(1983):642.

Reinhart G. Kussat

ADAPTING A VOCABULARY DEVELOPMENT PROGRAM FOR USE ON THE MICROCOMPUTER

The increased availability of microcomputers in audiovisual learning centers has made it possible to incorporate computers into a multimedia mode of instruction. At Bronx Community College, for instance, there are presently 18 microcomputers available for student use. This paper describes how Wordcraft (1970), a commercially available audiovisual vocabulary development program, was adapted to include the microcomputer.

Wordcraft was chosen because of its many excellent features: the program contains words at three levels of difficulty ranging approximately from grade 6 to 12 (Wordcraft I contains 300 words, and Wordcraft II and III contain 200 words each); the vocabulary is displayed on filmstrips accompanied by audio cassettes and is presented in context. Ten words together comprise a coherent narrative; the contextual narratives present worthwhile background information about important people and places. The teacher's edition contains practice exercises in a multiple choice, sentence completion, and matching format, as well as mastery tests.

Initially, mastery tests were programed for the microcomputer for each fifty words. A simple multiple-choice format was used requiring students to select a synonym for the target word from a choice of four words. If students choose correctly, the second command is to type in the target word; if they choose incorrectly, the right choice is provided for them, followed by the command to type in the target word. In this way, the process of testing becomes a teaching situation as well. A management system keeps track of students' responses and provides them with a printout of their score immediately following the test. A copy of the printout goes to the instructor, along with a class profile listing all fifty words for a particular test and indicating a frequency count of errors for each word. This system provides the instructor with information about each student's achievement as well as a complete class profile.

In this format, the vocabulary develop[illegible] program was used by basic skills classes for several semesters. Each we[illegible], fifty words were assigned as homework. Students were responsible for viewing the filmstrips and listening to the cassette tapes, as well as working the practice exercises which had been duplicated and made available to the students. Since students went to the Learn-

---

Reinhart G. Kussat, after completing his education at Rutgers University, taught developmental reading there at Livingston College and also at Bronx Community College. He is currently Director of Testing at Borough of Manhattan Community College.

ing Center at their own convenience, it was possible for them to work with the audiovisual presentation in a completely individualized way: they could stop the tape or filmstrip at any time, listen to pronunciation of words again, take notes, etc. The answers to the practice exercises were posted, so that students could prepare themselves completely before taking the final test on the microcomputer.

The next phase of the vocabulary development program began when the Learning Center acquired Microquest (MECC 1980), a "pseudo" authoring program. Microquest is a multiple-choice program which allows for four lines of information, and four choices with one correct answer, and with content matter to be chosen by the instructor. Used in the context of the vocabulary program, this format made it possible to computerize the practice exercises; students could then choose between the computerized version and the "paper-and-pencil" version of these exercises. In addition, pretests for each level of Wordcraft were prepared, making it possible to place students within the program more accurately.

The complete program has been used by twelve sections of the developmental reading course during the Spring 1983 semester and has provided the students with several beneficial experiences. Above all, the students seemed to become more independent learners; they began to schedule their time for using the audiovisual materials as well as the microcomputers. They learned how to operate and synchronize filmstrip viewers and cassette players; they became familiar with some basic operations on the computer; their elementary typing skills improved; and, since the computer requires exact information, they became aware of the need to type carefully and then to proofread their work. Once the students overcame their initial fear of working with the equipment, they enjoyed using it, particularly the computers. Finally, since the program was assigned as homework, it freed up valuable classroom time which instructors could devote to more individualized vocabulary work as well as other skills.

REFERENCES

Microquest. MECC Teacher Utilities Minnesota School District: Minnesota Educational Computing Consortium, 1980. Available from Computer & Information Science Unit, New York City Board of Education, 131 Livingston St., Rm. 408, Brooklyn, NY 11201.

Wordcraft. Wilton, CT 06897: Communicad: The Communications Academy, 1970.

Alice F. Randall

COLLEGE READING PROGRAMS: INSTRUCTORS CAN DESIGN THEIR OWN

To judge by the excitement induced by workshops in microcomputers for college reading instruction, a positive and pervasive change in instruction is imminent. The appearance is, unfortunately, partly deceptive. Many participants seem to be motivated less by a desire to learn the strategic and adaptive use of computer technology than by a yearning for a ready-made instrument to implant reading skills.

As a panacea for the limited time and energies of instructors in fulfilling their responsibility to large numbers of students, commercial software seems to be a godsend: a supplementary brain and persuasive teacher, reliable and trustworthy. Programs are available in a wide range of reading skill titles: vocabulary, recall, following directions, analogies, visualizing, charts and graphs, finding transitions, decision-making, and critical understanding. It is no derogation of some of the excellent commercial software to hedge their promise with qualifications. Commercial computer programs, like published textbooks, recordings, and videotapes, are supplementary aids to the instructor's own special talents in guiding effective reading and study. Of the many criteria for the selection of software, the most important is the degree to which it meets the needs of instructors and their particular students. No mass-produced instrument can fully meet this criterion.

Persuaded by the perceived needs of different student groups and aware of the relatively small size of the college market for publishers, Joseph Bourque (College English, January 1983) has concluded that "Ultimately, those who wish to use computers for instruction in English in higher education generally have to write their own software."

Reported in the literature are numerous examples of programs devised and used by college faculty for their own special populations. Programs range from the simple to the sophisticated microcomputer system devised by DeLuca (1980) for recording and measuring logical thinking in both physical and social scientific content.

Many reading instructors perceive reading instruction not as general skill instruction but as guiding students to cope with strategic demands of the various passages they encounter. Those whose theoretical base precludes extended

---

Alice F. Randall, Associate Professor in English, Christopher Newport College (Virginia), is a reading specialist whose publications include Reading Laboratory and Reference Manual, and articles such as "A Computer Managed Reading Laboratory Program," Reading World, May 1972.

---

use of skill programs often develop their own software in response to their desire to guide students to approach each individual reading assignment as a new situation where they must appraise the demands of the reading task and accommodate their own schemata.

The following description is designed to illustrate the practicability, as well as the problems and pitfalls, of the development by a computer novice of a program to reinforce his or her instructional concept. The procedural steps below are adapted from the splendid guide provided by Stuart Crawford (1981).

1. **Form a team**, including both pedagogical and computer specialists. In our case, the team consisted of the one full-time reading instructor in an urban four-year college and a student assistant--a senior majoring in computer science.

2. **Formulate the objectives**. In the author's opinion, at least the first attempt in computer programing should be an obvious extension of the instructor's theory of the reading process and his or her own teaching methods. This enables instructors to focus their attention on the computer capabilities rather than the niceties of their theoretical concepts of reading. Further, it shortens and simplifies revision and field testing later on.

In the present case, the objective was to produce an instrument that would reinforce a fundamental step in interpreting a nonfiction passage. Class instruction emphasizes a sequence of reading, identifying the principal argument, organizing the passage through diagram or outline, and writing and editing a summary. The summarizing task impels the student to focus on the central argument and supporting evidence and/or reasoning. In one stage of editing the summary, the student is taught to check the modes of exposition used in the original passage, as they offer clues to the author's objectives and organization. The computer program should lead students to: identify important ideas through a survey of the major modes of exposition the writer might use to express those ideas; check their summaries to see whether they have included them and, if necessary, revise their summaries to incorporate the additional ideas they have located.

3. **Analyze the task**. Decide the optimum form (drill and practice, tutorial, dialogue, simulation, or game type), the difficulty level, and the desired feedback for the program. In the present case, it was decided that a comparison of the student's summary with the original passage should be guided by the tutorial program mode, at the difficulty level of the student's textbook, with feedback primarily in the form of revision by the student of the summary on his or her own paper.

4. **Construct a flowchart**, with detailed instructions about frame design. In the present case, this step recommended by Crawford was passed over, to the author's later chagrin when the program demanded more changes than should have been necessary. The author simply wrote four double-spaced pages of copy in paragraph form, directing students to search for five modes of exposition: description, cause/effect, comparison/contrast, definition, and examples.

5. **Examine the match between the design and implementation.** The author gave a nod to Crawford's recommended step by asking a colleague who is a writing specialist to review the typed narrative copy. His suggestions, in the form of expanded explanations of the modes of exposition, proved to be of fundamental importance in the use of the program.

6. **Code the program.** The student assistant coded the program in about four hours, the only necessary equipment being an Apple II Plus micro, disk drives, and a printer. The coder has the opportunity for considerable ingenuity at this stage, for a variety of devices are available: flashing signals, graphics, color, and sound, depending on the capabilities of the available computer. The coder can work into the program instructors' special requirements such as the one Crawford cites of shutting down the program if the student replies to its questions with profanity.

7. **Document the program.** Explain why certain techniques were employed and develop support materials. In the present case, the coder kept a record of the procedures used, with supporting rationale. No supplementary pedagogical materials were deemed necessary.

8. **Field test.** The author selected volunteers from the class who were articulate and relaxed. They had written summaries with major omissions that could quite clearly be discovered by identifying the writer's use of one of the expository modes to emphasize his or her point.

Anecdotal evidence from the testing indicated that the program held promise. Testing also revealed deficiencies in the program which seemed obvious to the author as soon as they were directed to her attention, and the program went through successive stages, with some revisions still to be realized.

9. **Revise**...an endless process. Unlike a printed book, the program can be revised easily and frequently. The first revision of this program was a change in framing, so that the student could not proceed after reading one instruction without changing the frame.

The major changes still to be made are based on field trials:

1. Branching will be written so that, after being exposed to a short explanation of one of the modes of exposition, the student will be able to (a) proceed, (b) review what he has just read, or (c) read a more detailed description.

2. The program will be adapted for use with a student's diagram or outline of the passage rather than the completed summary. This will enable the student to concentrate on the information and ideas of the passage before attempting to phrase his or her summary in conventional form.

This program, directly related to the instruction in the developmental reading course, supplants classroom instruction only after principal instruction has been given and the student has reached the point of summarizing essays with a degree of independence.

The author has concluded, through preliminary success with the use of this program, that development of in-house software to reinforce instruction is within the capacities of even the computer novice. The prerequisites are conducting a literature review to spur his or her own creative talents, and finding a coder who possesses a reasonable amount of flexibility and creativity.

REFERENCES

Bourque, Joseph H. "Understanding and Evaluation: The Humanist as Computer Specialist." College English, 45 1(January 1983):67-73.

Crawford, Stuart. A Standard Guide for the Authoring of Instructional Software. Reference Manual Vol. 3. Victoria, B.C.: JEM Research, 1981. ERIC document ED 211 062.

DeLuca, Frederick P. "Measurement and Analysis of Logical Thinking." Final technical report. Ames: Iowa State University of Science and Technology, 1980. ERIC document ED 216 863.

Gerard M. Dalgish

ESL COMPUTER-ASSISTED RESEARCH AND INDIVIDUALIZED CAI

This paper describes a research project at Baruch College in which writing errors of ESL students are being entered, grouped, and analyzed by computer to determine the most common, cross-linguistic grammatical errors among first-language groups. Contrastive linguistic analyses of these grammatical categories will then be prepared, comparing the first-language structure with the corresponding English pattern to create language-specific, individualized grammatical lessons, to be available first in print and later in a computer-assisted instruction (CAI) program.

The idea for this ESL research came from a similar computer-assisted linguistic research project I worked on in 1978, on the University of Illinois' PLATO system. Some 700 sentences, from five East African languages, were entered on PLATO, with morpheme-by-morpheme analyses, a translation, and a set of "attributes"--in this case syntactic categories such as [declarative sentence], [1st person singular subject], [morpheme aa], etc. PLATO could be queried to list all sentences with specific attributes. By querying the computer appropriately, cross-listings of data were easily obtainable, and an analysis practically presented itself.

The same approach is now being used in Baruch's ESL writing research. The corpus here is an expected 5,000 sentences taken from in-class writing samples and from the CUNY Writing Assessment Test. These are entered on the VM/370 CUNY computer, utilizing the FOCUS interactive language data base system available from Information Builders, downloaded on IBM PC's. The languages--ranging from Arabic and Chinese to Russian, Spanish, Vietnamese and Yoruba--have sentence totals roughly consonant with ESL population breakdowns at Baruch. The error types, articles, determiners, vocabulary/idiom, prepositions, part-of-speech misuses, and others are included.

We can then ask: "Which are the most common errors found among ESL students?" and "What are the most common errors for each language?" Further refinements will answer such questions as: "Within a given grammatical error type, are there patterns peculiar to a certain language?" Some of these questions will have to await an increase in the corpus. Some totals of numerical searches by language are listed in Table 1 below.

Gerard M. Dalgish, Assistant Professor of English, Baruch College, has taught linguistics, ESL, and African languages at Dar es Salaam, Tanzania; University of Illinois; and Lehman College.

Table 1

Language	Total Number of Sentences	Article Determiner or Mass-Count Mistakes	Number of Vocabulary or Idiom or Preposition Mistakes	Number of Past Participle Mistakes	Number of Parts-of-Speech Mistakes
Arabic	11	2	8	1	1
Chinese	506	137	170	31	32
Farsi	19	11	7	2	3
French	15	5	6	3	1
Greek	104	30	31	2	10
Japanese	14	6	3	1	1
Korean	116	48	38	8	7
Russian	187	84	51	2	10
Spanish	221	37	73	18	16
Vietmanese	11	5	7	0	2
Yoruba	11	4	1	0	2

Some of the results are of interest to ESL teachers. The most numerous errors were in vocabulary and idiom and preposition use. These, to some extent, are not strictly grammatical errors, yet are not usually emphasized enough in ESL writing classes. Another extremely important category involves the article/determiner/number noun system of English--the hardest grammatical category for our students. Other error types and their totals help us determine what classroom emphasis should be.

The next step, the contrastive linguistic analyses comparing the error-prone English grammatical category with the student's first language, will have to wait until the corpus is complete. Preliminary analyses of sentences in which BE and HAVE, for example, have been confused, along with the expletive THERE, suggest that this confusion appears most frequently among Chinese-speaking students. The contrastive analyses will contain first a discussion of THERE and then a language-specific discussion. Contrastive lessons for other grammatical areas will follow.

In the meantime, we distribute sheets listing common errors, as proofreading exercises. Students correct their own language errors. Many have commented on the usefulness of this work, claiming they recognize "their" kind of error, an

indication of the psychological reality of the existence of error types specific to each language.

Individualized CAI is the long-range goal of the present research. We hope first to print and publish some of the cross-linguistic analyses and sample sentences in a proposed "Guide to ESL Writing Errors," and to take the research and findings and write ESL CAI programs, parts of which could be individualized by the native language of the student. We will know not only which general areas need most attention, but within each grammatical category which are the likeliest trouble spots for a particular language. I think that we can go beyond the realm of mere stimulus-response, fill-in-the-blank exercises with more imaginative use of movable graphics to show, for example, distinctions in verb tenses or aspects.

This is the direction I plan to take in CAI software. I hope that publishers and other ESL educators will go along with it. The software publishers, programmers, and teacher-authors must form a consortium to hasten the proliferation of strong ESL software, and ultimately to give much-needed impetus to the market for CAI.

Clara Velazquez,
Elie Lowie, and
Juan Vega

AN ESL/COMPUTER-BASED PROJECT

This paper describes a curricular model which can provide students who demonstrate only limited proficiency in English the opportunity to receive introductory instruction in the area of computer technology. Developed in cooperation with CUNY's Office of Special Programs, the model has as its immediate goal the exposing of intermediate and/or advanced ESL students to an introductory course on computers, highlighting such topics as computer literacy, word processing, and elements of data processing or computer science. The long-range goal is to introduce these students to career opportunities in computer-related fields, while at the same time accelerating their acquisition of English.

In this model, students enroll in a program consisting of the beginners' computer course, an ESL course and a tutorial laboratory. A computer course might meet, for example, two class periods a week, with the ESL instruction and the tutorial sessions together taking perhaps an additional three class periods. In the bilingual tutorial laboratory, the instructor explains, using the student's native language, the main concepts to be covered in the next meeting of the computer class. The ESL teacher is provided with a list of vocabulary (with appropriate explanations) culled from upcoming computer lessons in textbook chapters. Students are familiarized with this vocabulary at least one week prior to its use in the computer class, so as to feel comfortable with the new material when it is covered entirely in English.

The computer course focuses on the use of microcomputers and includes an introduction to the programing language, LOGO. Students learn to write programs of particular interest, such as one to calculate a student's tuition payments. Pairs of students work at computer stations each of which consists of a micro, a data storage device (such as a disk drive), and a printer. Topics are selected from the data processing and the introductory computer science curriculum. A

Clara Velazquez is Professor of ESL, Department of English, Hostos Community College. Besides ESL, her expertise covers teacher-training, and curriculum development and evaluation.

Elie Lowie is Professor of Physics, Queensborough Community College, where he chairs his department.

Juan Vega, now a teacher of computer science in Puerto Rico's higher education system, was formerly coordinator of the New York University Computer Laboratory at Purchase.

---

sample syllabus is available from the Office of Special Programs, CUNY, 500 West 57 Street, 16th Floor, New York, NY 10019.

In addition, a career guidance counselor is available to explain to the students the various career options in computer-related fields, to nurture their self-concept as well as to motivate them to reach higher levels of achievement in what clearly are marketable skills. It is hoped that all these efforts will reinforce each other, enabling students to feel competent in the subject matter as well as to think and to communicate in the English language.

Henry Africk

HOW MICROCOMPUTERS CAN BE USED TO HELP TEACH BASIC MATHEMATICAL SKILLS

At New York City Technical College we have been using computers to teach basic skills in mathematics since 1980, when we received a National Science Foundation grant to develop computer-assisted instruction programs for our math lab. Our experience with this innovative form of instruction led to the views expressed in this paper, which are those of the author and do not necessarily reflect NSF's views.

Computers as an Instructional Tool: Computers are best suited for doing simple repetitive tasks quickly and accurately. For instructional purposes, this means the teaching of skills involving short simple steps that can easily be checked but which require extensive student practice. Fortunately, most basic mathematical skills fall into this category. The computer can be programmed to generate endless drill-and-practice problems of a given format and to check answers to these problems. More sophisticated programs can also be written to identify common student errors and to provide help at the steps where the student has difficulty. An assignment to do problems on the computer can be an attractive alternative to a traditional homework assignment particularly if there is some way of recording what the student has done. At New York City Technical College we have the computer print out, at the end of each session, a summary of the student's work, which the student then gives to the instructor. (See Example 1 below.)

In this system a student cannot simply copy another's assignments since the problems which the computer generates are varied. Students try harder because they know that their work is being checked, and the instructor is relieved of the chore of grading homework assignments. Upon the student's request, the printer will print out all or part of the problem. Furthermore, the computer will print out, when requested, the complete solution to a problem, either one of those generated by the computer or one of a similar type that the student supplies himself. (See Example 2 below.) Thus students need not wait until the next class meeting or seek other assistance to find the answers to problems they have had difficulty solving.

The computer can also be used for individualized, self-paced, or programed instruction. This is especially convenient when used in conjunction with a

---

Henry Africk, Associate Professor of Mathematics, New York City Technical College, is Director of a National Science Foundation CAUSE project for improving student progress in mathematics by using computer-assisted instruction.

---

workbook. However, it should be kept in mind that the computer can only respond with a limited number of preprogramed responses, so it cannot truly replace traditional classroom lectures and discussion. Computerized lessons are not replacements for textbooks either. Computer screens are harder to read than books and obviously not as portable. It is also much more difficult to write instructional computer programs than comparable textbook material. Finally, it is unlikely that suitable instructional programs will be available for all the topics in the curriculum.

There are other ways that microcomputers can help with mathematics instruction. A microcomputer with a large television screen can be brought into the classroom to display graphs and do complex computations. For example, our instructors find it much easier and better to have the computer display the graph of an equation in polar coordinates than to draw it themselves on the blackboard. There are educators who advocate students be taught how to program the computer themselves to solve the mathematical problems they are studying. However, students struggling to master basic mathematical skills are probably unsuitable candidates for this approach, since learning to program a computer is difficult enough in itself.

Hardware and Software: Unless you plan to develop all of your own software from scratch, your choice of a computer will depend on what instructional programs already exist for it. Not only will you want commercial and noncommercial instructional programs that can be used immediately, you will want to see how others have overcome various technical problems particular to a given machine should you decide to write your own programs. At the moment, the Apple II dominates the educational market because of the large amount of educational software available for it, though this may change as better and cheaper machines are introduced. Even for the Apple, you will probably find suitable programs for only a few of the topics of the basic mathematics curriculum. This is because the difficulty of writing educational programs and the ease of illegally copying them have made educational software publishing relatively unprofitable.

It is also necessary that style and content of an instructional program closely match the way the topic is taught at your institution. Unlike a textbook, it is not usually practical to add supplemental material to an existing computer program or to tell students to ignore a part of the program which you do not feel is appropriate. It is for these reasons that at New York City Technical College we decided to write our own programs. However, because of the amount of time required to develop each program, we have covered, after three years' work, only a small part of the basic algebra curriculum. Most colleges will therefore want to begin by acquiring software developed by others. The best way to find out what is available is to attend meetings of professional organizations interested in using computers in mathematics education, such as the Mathematical Association of America, the National Council of Teachers of Mathematics, and the National Educational Computing Conference.

Implementation: There are essentially two locations which can house a system of computer-assisted instruction: the classroom and the math lab or learning center. Most colleges will probably choose the math lab since there is usually not enough time during regular classroom sessions for both lecture and computer

practice. Also specially equipped classrooms, with enough computers for each student, are required if computer work is to be part of regular classroom instruction. The math lab offers more flexibility since not all the students need to use the computers at the same time. Still, there must be some way of checking that students actually go to the math lab and do the problems. Our experience is that students generally will not make full use of computer-assisted instruction facilities unless they are required to do so. Our solution, as mentioned above, is to have a computer printout of the student's work which the student then gives to the instructor. Another possibility is to network the microcomputers to a central storage device which can store a large number of records. The choice of the kind of system to use will depend on the funds available and the number of students to be served. Colleges having limited funds will probably begin with a simple system consisting of just a few microcomputers and printers situated in a math lab and serving a small selected group of students.

Conclusion: How effective is computer-assisted instruction for mathematics remediation? The answer depends, of course, on many factors, including the quality of the instructional programs, their suitability for the student population, and the manner in which the project is implemented. In our case, the sections of our elementary algebra-geometry course which were given computer assignments did about 10% better than other students on the most recent departmental final exam. We plan further studies of such examination scores to see how well these results hold up over time and how we can do better. Our computer-assisted instruction programs appear to be most helpful to students of average ability who just need more practice to master required skills. The computer assignments make less of a difference for better students, who will master the work in any case, and for the least able students, who will still need additional help. Aside from improving their performance, most students do find the computer programs an easier way of learning than just doing textbook problems. We also get fewer complaints from students about not being able to understand the course topics that are covered by computer assignments.

To summarize our experience, there is still much to be learned about how computers can affect the learning process. This is a new and exciting educational medium that allows us to innovate ideas and solutions to traditional educational problems. The chief obstacles are the shortage of good instructional software and the lack of adequate institutional funds to support microcomputer projects.

Example 1

[This example is from our "Plotting Points and Graphing Lines" program. The summary gives the number of errors in solving for X and Y, the number of requests for help, and the number of errors in plotting the points.]

HENRY AFRICK — GRAPHING LINES
MAY 1, 1983 — TIME IN: 9:30

(continued on next page)

SUMMARY OF YOUR WORK

PROBLEM	TYPE	SOLV ERRS	HELP SOLV	PLOT ERRS
X + Y = 6	1	0	0	0
X + 2Y = -4	2	7	7	1
3X - Y = 6	2	0	0	3
X - 2Y = -4	2	4	4	0
Y = 2Y + 5	3	1	1	1
X = 2	4	0	0	0
Y = 3	4	0	0	0

TOTAL NUMBER OF PROBLEMS: 7 TIME OUT: 10:00

Example 2

[From our Plotting Points and Graphic Lines program]

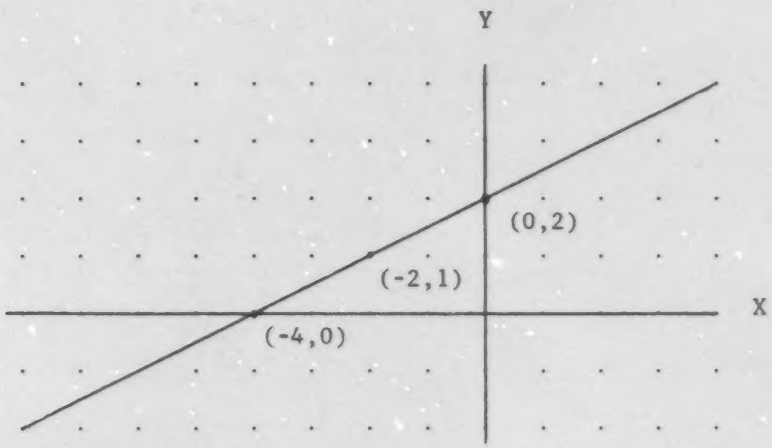
Graph the equation X - 2Y = -4:

Let X = -2:

```
 X - 2Y = -4
-2 - 2Y = -4
   -2Y = -4 + 2
   -2Y = -2
     Y = 1
```

X	Y
0	2
-4	0
-2	1

The third point is (-2,1).



X	Y
0	2
-4	0
-2	1

The graph of X - 2Y = -4.

John C. Miller

DESCARTES' DELIGHT: A LANGUAGE FOR COMPUTER-INSPIRED LEARNING IN MATHEMATICS

Most students in mathematics classes at all levels lack the background needed to understand the new material being taught on any given day. They also lack the necessary experience in working with examples and concrete applications of the mathematical abstractions being taught in any given class. Under these conditions, students tend to learn mathematics as a system of arbitrary rules, devoid of real content and meaning. For example, it makes no sense to teach addition to students who lack experience in counting sets of objects, combining them, and counting the union; or to teach signed arithmetic to students who lack experience in using a number line to locate and measure directed quantities and to combine them graphically; or to teach algebraic rules to students who lack background in substituting numbers for variables and computing and graphing the results.

Students forced to work with mathematical concepts without having the necessary prerequisites predictably tend to adopt unproductive and ultimately self-defeating learning strategies. They memorize rules and procedures without comprehension and thus can only apply them correctly, if at all, to essentially identical problems. They grasp at shortcuts which apply only to special cases. They concoct "solutions" consisting of pieces of solutions to other problems they have seen. Worst of all, once a student starts this kind of groping approach to mathematics, it is all but impossible to get him back on the right track.

In recent years, there has been increasing awareness, especially in the lower grades, of the need for practical, hands-on experience with mathematical concepts before abstract principles are taught. Number games, opportunities to explore mathematical concepts via manipulation of sets of objects, and geometrical explorations using objects of various shapes, are more common than formerly in elementary schools across the country. For slightly older students, however, much more effort in this direction is needed.

Uses of Computers

Most attempts to apply computers in mathematics instruction have followed the standard classroom pattern, drilling the user in skills without sufficient con-

---

John C. Miller, Associate Professor of Mathematics at The City College of New York, is currently writing computer programs for instruction in elementary algebra.

---

cern for the prerequisite experience in applying the principles to real objects and situations. Some instructional programs, typically utilizing computer simulations, attempt to provide such experience and these are generally among the most effective and interesting.

Another approach to using computers in mathematics learning is to encourage students to become programmers. This is particularly promising because "teaching" a computer to do something is a very effective way to expand one's own understanding of the subject, and the errors or "bugs" which inevitably result are the most instructive part of the process. The best-known example of this approach is the LOGO programing language, developed by Seymour Papert and his associates at M.I.T., which encourages a student to explore physical and geometrical concepts in the course of writing computer programs to produce visually interesting output on the computer screen.

The Descartes' Delight Language

A new programing language, called Descartes' Delight, extends this approach to mathematics readiness by utilizing standard mathematical terminology and notation in its fundamental vocabulary and syntax, and helps the user to obtain a working knowledge of substantial portions of algebra, analytical geometry, and pre-calculus in the course of programing the computer to generate a variety of complex graphic effects. Mathematical discovery becomes irresistible.

An elementary Descartes' Delight program:

The user types:	Comments:
COMMAND TENT (HEIGHT) PLOT (-1,0) JOIN (0,HEIGHT) JOIN (1,0)	A new command called TENT is added to the language. A TENT consists of two straight line segments, with HEIGHT as a parameter.
COMMAND DIAMOND (SIZE) TENT (SIZE) TENT (-SIZE)	A DIAMOND consists of two TENTS, one of which has a negative height.
COMMAND WING DIAMOND (1) DIAMOND (2)	A WING consists of two superimposed diamonds, of different sizes.
COMMAND ANTENNAE SCALE (.2,1) TENT (-2) SCALE (1,1)	ANTENNAE are formed from an inverted tent, with temporarily compressed X-axis.
COMMAND BUTTERFLY ORIGIN (-1,0)	A BUTTERFLY consists of two wings and a set of antennae, located

(continued on next page)

```
WING                          by translating the origin of the
ORIGIN (2,0)                  coordinate system.
WING
ORIGIN (-1,2)
ANTENNAE
ORIGIN (0,-2)

BUTTERFLY                     This command draws our butterfly
                              (see Figure 1 below).
```

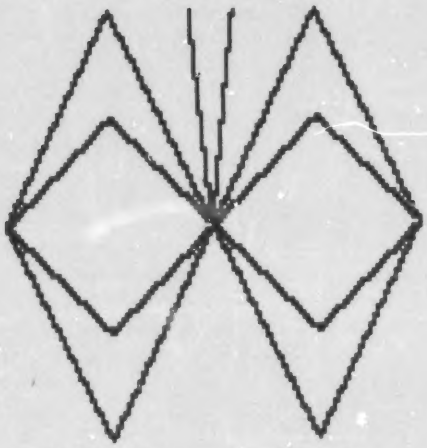


Figure 1. BUTTERFLY

Butterflies can now be located anywhere on the screen, and rotated to any angle, using coordinate transformations available in Descartes' Delight. With the kaleidoscope command, the screen can be filled with flying butterflies.

A more sophisticated program uses a sine curve to produce a "flower":

```
User types:                   Comments:

COMMAND CALCSINE (H)          A new command called CALCSINE is
DOMAIN X=-1,1                 added to the language. CALCSINE
INTERVALS (30)                computes a table of values for a
FUNCTION Y=HSIN(3.14X)        one-period sine function, with
                              amplitude H.

COMMAND FLOWER (SIZE)         A new command called FLOWER is
CALCSINE (1/3.14)             defined, which combines four sine
SCALE (SIZE, SIZE)            curves to form a flower. The axes
GRAPH (X,Y)                   are rescaled using the parameter
SCALE (SIZE,-SIZE)            SIZE, and the vertical axis is
```

(continued on next page)

```
GRAPH (X,Y)
GRAPH (Y,X)
SCALE (SIZE, SIZE)
GRAPH (Y,X)
```

temporarily inverted. Later, the axes are interchanged and the inverse sine function is graphed.

```
FLOWER (1)
DANCE (2,2)
```

A FLOWER is drawn, and then the screen is filled with a moving kaleidoscope display of flowers (see Figures 2 and 3 below).

Figure 2. FLOWER

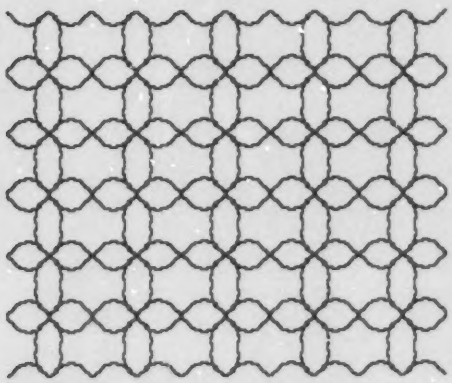


Figure 3. DANCE

The most spectacular feature of the language is the kaleidoscopic graphics capability summoned by the DANCE command in the preceding program. Any graphic output may become the input to the kaleidoscope routine. Its seven parameters (selected randomly if the user opts not to supply them all) allow a stunning variety of moving visual patterns to be generated. This feature is designed to inspire the user to explore various means of generating graphs as input to the kaleidoscope routine, and thereby to acquire a broad range of mathematical experience and skills as he writes the needed programs.

Parametrically defined functions may be graphed as easily as simple functions by specifying a domain variable distinct from the variables to be graphed. Although primarily intended for graphics use, the language is also a general-purpose, structured computer language. For example, number-guessing games and entertaining simulations can easily be written. Using the language produces a working knowledge of structured programing, an understanding of both global and local parameters, and a sense of confidence on the part of the user that a computer is a useful and interesting tool. It includes an easy-to-use disk operating system for saving programs for use on a later occasion.

Students who explore such programs obtain a readiness for formal mathematics that others lacking this experience can not hope to match. They see the geometrical effects of negative numbers, and "know" what they are in a meaningful way. They understand coordinate systems, and changes of coordinates, through repeated use. These students use parameters and variables, and watch the effects of varying their values. They know such concepts as "origin," "function," and "domain" intimately, through direct use. They watch the computer generate tables of values for functions, and see the values being translated into the graphs of those functions. Such students are prepared, as few students today ever are, to learn formal mathematics.

Alfred Bork

BASIC MATHEMATICS IN COLLEGES AND UNIVERSITIES: COMPUTERS AS A SOLUTION

Universities are typically poor in dealing with their students' heterogeneous background in basic math skills. The problems of underprepared students are seldom those that can be handled well in a "whole remedial course" procedure since each one has unique needs. While we give lip service to treating students individually, very little of this actually happens in our classes.

As a result, "learning resource centers"--inner universities--have arisen on many campuses in recent years. These are courts of last refuge for students in trouble in their courses. The existence of learning resource centers is a strong condemnation of our traditional courses. If we were doing our job fully, such centers would not exist.

I will argue that a clear way of dealing with this situation exists which, if carried out nationally, could solve the problem almost everywhere in the country and at a reasonable cost. I refer to the development of flexible, highly interactive, graphic, computer-based learning modules to teach elementary basic mathematics. This material would be used by students strictly on their own, or it could be available in learning resource centers.

In those universities, such as Carnegie-Mellon University, moving toward requiring all students to have computers, the material would naturally run on those computers. We would expect it to be available on a range of the common computers coming into use in schools at present. No course structure is needed with this material as the units would be directly usable by all students. Some conventionally organized universities would use them within courses, too. The emphasis would be on mastery learning, demonstrating conclusively that the student fully knows each of the mathematical tasks involved.

The basic format of these materials is a series of problems, covering the entire spectrum from simple arithmetic through algebra and trigonometry. The problems are all presented at the computer. Each problem arises out of a computer-based problem generator so that an infinite number of similar problems are available to students. Students are able to seek immediate assistance in connection with each problem; this assistance is highly interactive computer aid, not just verbal information which the student reads passively. Assistance is

---

Alfred Bork is Professor of Information and Computer Science, and Director of the Educational Technology Center, University of California, Irvine. His recent books are Learning with Computers (1981) and Personal Computers for Education, the latter being published by Harper & Row in 1984.

---

also available on a broader scale for students who are having severe difficulty with certain identifiable types of problems. They are then recycled through the exercises, with additional aid if necessary, until they can perform at a mastery level on each type of problem.

Note that in material of this kind, students move very quickly over anything they know; student effort is focused exactly on the material they need help with. The computer materials are also flexible; particular problems are taken out of the pool, or other changes made in the way the exams run, to suit individual circumstances.

The materials follow the best strategies available for producing modern computer-based learning materials (Bork 1981). Extremely competent teachers from all over the country can be involved in the pedagogical design sessions to structure the material (Bork, forthcoming).

Although no careful cost estimates have been made, comparisons of this project with other efforts at producing very high quality learning material on the computer suggest a budget of about $350,000; two to three years would be required. It is expensive to produce good curriculum material, independent of the medium. We could not expect these materials to be produced by one teacher in that teacher's spare time.

Whether we regard these costs as considerable or as small depends on the scale being considered. That is, if the materials were used in only one or two universities, the cost would be great. If the materials were available throughout the entire United States or even abroad, then the cost per school would be small.

A number of sources of funding are possible:

Federal Grant: This would be the simplest way to fund a project of this kind.

Commercial Support: Funding from a commercial publisher or computer vendor, with the idea of marketing the material.

University Consortium: A group of universities working together, each contributing some funds to the overall project. Presumably all would then have free use of the material, except for duplication costs, and could market it to other institutions, perhaps making a profit eventually.

Combination Funding: Although it is complicated, funding might come through some combination of the above sources.

In conclusion, adequate material of this kind could provide most American universities with an extremely flexible way of meeting the needs of mathematically underprepared students and would conserve valuable student time. I frankly can see no better alternative.

REFERENCES

Alfred Bork. Learning with Computers. Billerica, MA: Digital Press, 1981.

Alfred Bork. "Production Systems for Computer Based Learning." Forthcoming.

Elwyn H. Davis

SOME MATHEMATICS SOFTWARE I HAVE KNOWN--AND HOPE TO KNOW

Recently the Journal of Mathematics and Computer Education instituted a software review column. Now nearing the end of my second year as an editor of that column, I have reviewed a reasonably large variety of software aimed at students in the first two years of college mathematics courses and am concerned with the lack of commercially available material. Most of what has come to our attention, which is aimed at mathematics instruction, has been of good quality; however, there is a great and glaring gap between what is available and what would be desirable.

This situation perhaps results from the lack of a perceived market on the part of publishers. There is no doubt much more incentive to prepare materials for high schools and elementary schools than for colleges, since the sales possibilities are so much greater there. Perhaps as more publishers become interested in educational software, the picture will change. There is certainly no dearth of talent among mathematics instructors. I have seen some excellent products from teachers which have not been marketed. Hopefully, as the number of microcomputers in colleges expands, the market will look more favorable.

But let me proceed to my two major points: First, I will give a brief survey of submissions to our journal which were directly related to pre-calculus instruction. Then I will discuss directions in software development to be hoped for in the near future.

A large effort in algebra tutorials has been undertaken by EduWare Services, Inc., Box 22222, Agoura, CA 91301. This company has developed a six-disk set of materials to cover algebra from the beginning through quadratic equations and inequalities (Algebra 1, 1981; 2, 3, 1982; 4, 1983; $39.95 ea.; 5 and 6, 1983, $49.95 for both). This series of disks should not be considered a replacement for textbooks. It is advertised as a tutorial product. In this it succeeds. While the series was probably developed for high school students, it can serve as a tutorial aid in colleges.

Conduit, Box 388, Iowa City, IA 52244, has produced several packages which could be used in pre-calculus mathematics instruction: Algebra Drill and Practice I and II (1982, $125); Discovery Learning in Trigonometry (1982, $75); Descriptive Statistics (1978, $75; student guides $5.95); Interpreting Graphs (1982, $45); and Arbplot, a versatile graphing package (1982, $125).

Elwyn H. Davis, Professor of Mathematics, Pittsburg State University (Kansas), was formerly with Pennsylvania State University and Phillips Petroleum Company.

Avant-Garde Creations, Box 30160, Eugene, OR 97403, a firm better known for games and graphics, has developed two testing packages for New South Wales, Australia schools using the powerful authoring Zenith Education System. While these tests are not aimed at the U.S. population, they could be useful. They are entitled Real Number System and Parabola and Quadratic (1979, $29.95 ea.).

Programs originally developed by Math Software in 1981, are now revised and published by D.C. Heath Co., 2700 N. Richardt Ave., Box 19309, Indianapolis, IN 46219. While dealing mostly with calculus, two could be useful in pre-calculus instruction. They are Function Grapher, which does what it says, and Solving Simultaneous Linear Systems. The latter could be quite helpful in the classroom as it relieves the user of tedious arithmetic in solving 2 by 2 systems. These programs are packaged together with other programs (revised 1984, $100 to $250 per package).

Edu Tech, 634 Commonwealth Ave., Newton Centre, MA 02159, markets two programs which could be useful in pre-calculus mathematics: One, Math Lab I-Graphs (1981, $39.95), meant for pre-college but useful at beginning college level, is a program which leads students through graphing straight lines and parabolas by using game paddles. Another program is Six Puzzles (1983, $39.95), which uses six classical puzzles to introduce students to problem solving using graph theory. It could be used for general education or liberal arts mathematics courses.

Now, to turn to some hoped-for developments. My three concerns are: (1) There is a great need for programs directly related to problem solving; (2) the curriculum and ways of teaching should be altered by the use of computers; and (3) the cost of software must come down.

Concern number one: The mathematical education community recently agreed on ten basic skills to be sought after in instruction. One of those was problem solving, which I believe to be the most important goal. Since most people study mathematics as a tool, problem solving is at the heart of all mathematics instruction. This area cries out for great effort to develop software. Of course, programs which relate to algebraic and arithmetic problems are vital to such an effort. But the computer opens up the possibility of solving many non-algebraic problems. In fact, many of the real-world problems faced by people are not amenable to algebraic solution. They are combinatorial in nature, requiring formal or informal enumerating of possibilities, choosing from those possibilities, or ordering the possibilities. Such problems are often tedious, but the computer puts their solutions within reach.

As far as algebraic and arithmetic problems are concerned, there are several directions that software developers might take to aid such mathematical instruction. First, and perhaps easiest, there is a need for programs which treat traditional methods of solving written problems, much as found in textbooks but using the computer's capabilities. Some of the authoring languages in existence today may be of assistance in this effort. Second, the capabilities for dynamic graphic displays should be used to aid students in visualizing problems. Third, the computer can be used to reduce the often bewildering degrees of freedom which students have when solving problems. Visual displays can call for parts of the required data or parts of formulations, and immediate feedback can

then be given. A step-by-step increase in the amount of autonomy given the student can be administered by the computer. Further exploration of the capabilities of the computer to aid in problem solving instruction is needed.

Concern number two: The ability of computers to open up new areas of problem solving should have an impact upon the curriculum. An example of a nonalgebraic problem given to my son's fifth grade class demonstrates this point. The problem given was to determine how many pennies, nickels, dimes, and quarters (with certain constraints), could be used to make a certain amount of money. The group found a solution, and the teacher was satisfied. I queried my son later about whether they had found <u>all</u> solutions. This concept had not been mentioned in class. We wrote a short program to try out all valid combinations of coins, and found that there was indeed only one solution. There are many such questions one encounters, and elementary arithmetic and algebraic techniques do not suffice to solve them.

This example indicates how a change in emphasis in problem solving, or instruction, is easily made to encompass types of questions solvable on the computer. Such changes will require much more emphasis upon algorithmic and combinatorial thinking than is currently present in our courses. The computer can make the same type of significant impact upon education that it has upon industry and science if we, as educators, will allow it to do so. In those fields, the computer is an organizer, calculator, searcher, and problem generator. In fact, as is well known, some scientific areas have been revolutionized by the capabilities offered by the computer. Some innovative teachers have altered their courses to take advantage of these capabilities, but as yet commercial software products have not appeared which really aid the needed change in curriculum.

My third and final concern: Software just costs too much!

In conclusion, let us hope that the next few years will see accelerated development of software aimed at problem solving and that proper changes in curriculum will emerge, changes which utilize the unique capabilities of the computer to unveil better problem solving techniques.

Lawrence Sher

MICROCOMPUTER FEATURES FOR TEACHERS

For teachers, the main use of microcomputers is to provide better instruction than was previously possible. This article will illustrate three features which might help achieve this objective. These are immediate error correction, instant readout, and arcade-style paddles.

Immediate Error Correction

When students try to solve a complex problem involving a large number of steps, an early error can negate the learning value of subsequent work. If the purpose of the exercise is to have students draw conclusions from their results, negative learning may occur. In BASIC, the "IF ...GOTO" statement is the key to error correction. In a typical basic skills exercise, a student might be attempting to find the first digit of the quotient in the example:

$$47\overline{)1993}$$

The student might input a 2 as the first digit of the answer. The program step might read: 150 IF X%<4 GOTO 300. Step 300 would read: 300 PRINT "SORRY, TOO SMALL" or 300 PRINT "SORRY, THE CORRECT ANSWER IS 4" depending on how much information you want to give to the student.

Instant Readout

The student can, however, learn more from his or her error if the instant readout feature is used. You show the student the result of multiplying his or her guess by 47 with brief comments. On trying 2 as the first digit of the quotient, the student sees:

```
   2          SORRY, TOO SMALL
47)1993
  94
 105          REMAINDER TOO BIG
```

Lawrence Sher, Professor of Mathematics, Borough of Manhattan Community College, has developed interactive microcomputer programs to teach statistical concepts under a MISIP grant, and programs to teach business mathematics under a VEA grant.

Since the computer does the multiplication and subtraction, the student has no chance to make a second error to cancel the first, and can see that the remainder is the key to testing his or her choice of quotient.

Paddles and Joysticks

A paddle or joystick allows for students to make rapid changes on the screen with ease.

In our problem, students can use the paddle instead of an input statement to enter the first digit of the quotient. By moving the paddle, they can quickly test several possible quotients. Each user can experience trying 9 possible quotients in the time that it would take to do one on paper. On the Apple II computer, the paddle has 256 possible positions and each, in order, is assigned a number from 0 to 255. The step:

```
10 X% = PDL(0)/26
```

will give a result of 0 to 9 depending on the position of the paddle. The X% variable gives only the integral part of the result.

The following loop will allow you to change a digit from 0 to 9 in the center of the screen using the paddle:

```
10  FOR N = 1 TO 100
20  HOME
30  X% = PDL(0)26
40  VTAB 12
50  HTAB 20
60  PRINT  X%
70  NEXT   N
```

Notice that the HOME statement erases the number so another can be put in its place even if the number of digits should differ. The TAB statements position the digit.

Paddles could be used for nonmathematical purposes as well. A similar program could have students use one to test many possible adjectives in a sentence. Thus, these features--immediate error correction, instant readout, and paddle control--can allow a more active, inductive learning experience than was previously possible.

Jo Ann Comito, Meryle Kohn,
and Anna Silverstein

COMPUTER ENHANCED LEARNING IN ALGEBRA AND TRIGONOMETRY: AN ECLECTIC APPROACH

Fall 1983: John Q. Student, college freshman, grapples with algebra and trigonometry homework. He entered school in September equipped with an SAT mathematics score well below 500, some vaguely remembered or misremembered algebraic facts ("I thought two negatives make a positive"), and some clearly remembered traumatic experiences in math classes (leading to abject terror at the sight of the words "a twenty percent iodine solution is mixed with..."). John expects to graduate in 1987 with a degree in computer science.

Fall 1984: John is now enrolled in the first semester of a rigorous differential and integral calculus sequence. He is coping competently with such topics as epsilon-delta definitions of limits, the differentiation process, and maximum-minimum verbal problems as a result of much hard work and an innovative freshman-year mathematics course that

- stressed all math skills required for calculus
- gave him ample opportunity to develop and strengthen skills taught in the classroom through use of a computer in direct mode for computation and with specially selected programs for discovery learning, tutorial, and drill
- developed his problem solving ability by teaching him to program his own solutions to problems
- created a stimulating, challenging mathematics learning environment

An impossible dream or a typical success story?

The authors are at New York Institute of Technology. Jo Ann Comito is Senior Coursewriter for Title III at the Old Westbury campus; Meryle Kohn, Assistant Professor of Mathematics at the New York campus, is currently teaching and developing LOGO software for the course described in this paper; Anna Silverstein, Assistant Professor of Mathematics at Old Westbury, has also taught mathematics at LaGuardia Community College where she developed the mathematics component of the computer science program. Her research interests include symbolic logic and combinatorics.

A two-semester algebra and trigonometry sequence featuring computer-enhanced learning, being developed at New York Institute of Technology, is designed to make such success the norm rather than the exception.

Not all our entering students are quite as poorly prepared or have career aspirations quite as high as the hypothetical John Q. However, since the Institute caters primarily to engineering and technology rather than to liberal arts majors, virtually all remedial mathematics students will eventually take higher mathematics courses including some calculus. Thus they must acquire a solid pre-calculus background quickly.

The pilot course offered in September 1983, met five hours each week with approximately three hours devoted to classroom instruction and two hours to individual work in the computer laboratory. Class size was limited to 30 with at least one Franklin computer for every two students in the laboratory. A Franklin with large monitor was available for classroom use.

Eclectic use of the computer gave the course its five most innovative features:

1. The computer was used in the classroom whenever its graphic and computational capabilities made it an improvement over the blackboard.

2. Students learned to run prepared programs and to use the computer in direct mode for problem solving.

3. Elementary programing in LOGO was taught. LOGO is a procedural, interactive, and extendable language that can help develop problem-solving skills without requiring many hours of detailed programing instruction. (See more details below.)

4. New software developed for the course emphasizes teaching mathematics through gaming strategies. Some of the games, categorized as "looking ahead software," encourage discovery learning by focusing student attention on problems and concepts before they are introduced in class, and provide an interesting and motivating framework for drill and practice.

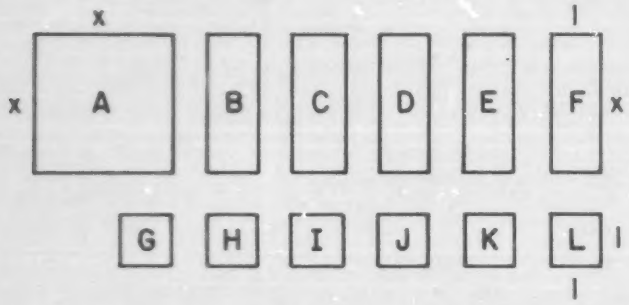
5. Commercially available programs identified as being of good quality were used by students in the laboratory for tutorials and for drill and practice.

Of the five uses of computers mentioned above, the two most interesting are LOGO and the development of our own in-house software.

Locally Developed Software

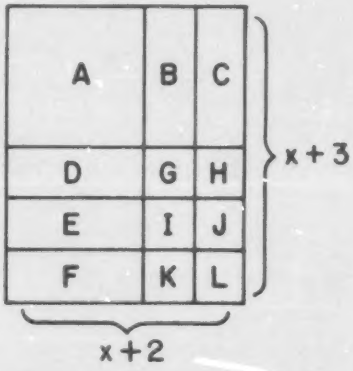
In our philosophy, educational software should make learning fun. We therefore have chosen to use game formats in developing software for the presentation of mathematical concepts. Below we describe three of our programs:

Bulldozer Factoring. This program falls into the category of "looking-ahead" software. It capitalizes on a geometric representation of trinomials in order to introduce the concept of factoring. The quantity X^2 is represented by a square with sides X units in length. Similarly, the quantity X is represented by a rectangle measuring 1 unit by X units and the quantity 1 by a square 1 unit on a side. Enough 1 by 1 squares, X by 1 rectangles, and X by X squares are drawn to represent the trinomial. (See Figure 1 below.) If the trinomial is factorable, then the squares and rectangles can be arranged to form a rectangle of unique dimensions. The lengths of the sides of the rectangle represent the factors of the trinomial. (See Figure 2 below.)



$x^2 + 5x + 6$

Figure 1.



$(x+2)(x+3) = x^2 + 5x + 6$

Figure 2.

The first time students use the program nothing is said about factoring trinomials. Instead, students witness an explosion at a rectangular storage yard. The storage yard flies into pieces and students must put the rectangle back together. They are supplied with a bulldozer for pushing the pieces around and are offered a $1,000 reward for rebuilding the original rectangle. They have up to three tanks of fuel for the bulldozer and pay for the fuel they use out of the reward money. After each tank is consumed, a hint is given about the rectangle that is to be formed. Any reward money left becomes the score. As an added incentive, the names of the five top scorers are saved on the disk.

The game format and spirit of competition encourage students to keep trying and to work out better strategies for determining the placement of the pieces to form the rectangle. After factoring is taught in class, the students again play the bulldozer game. Being able to factor the expression is a big advantage in playing the game. The greater a student's facility with factoring, the easier the game becomes mathematically and students can concentrate their skills on manipulating the bulldozer.

Trigs in Space. This second arcade-type game is used for drill and practice in trigonometric facts. Students control the rotation of a spaceship being attacked by aliens. The ship's radar provides students with a report on the angular position of the alien. This report is always accurate, but not immediately useful. When the report is given in degrees, the ship must be aimed in radians, and vice versa. At higher levels of play, the report is in terms of a trigono-

metric function of the angle. The aliens always appear at one of the key angles and the game is designed to help students remember these key relations rather than help them with the use of tables or calculators.

Students start with three ships and can earn additional ships by destroying enough alien vessels. The score for each hit varies inversely with the time needed to fire accurately. Knowing the relationships well and shooting quickly greatly improve the score. As with the Bulldozer program, the names of the top scorers are recorded on the disk.

Alge-Bomb. While still using a game format, Alge-Bomb is not an arcade-type game requiring quick recall and quicker reflexes. Rather it is a multilevel high resolution adventure that takes place in a top-secret laboratory. It is designed to help students solve various types of equations.

At the simplest level, students must solve three linear equations the solutions of which represent the combination to a safe. The safe contains a bomb which, if not defused, will explode, destroying all the documents and secret formulas within. Students can receive help from various laboratories in the building. For example, the linear level contains a graph lab, a multiplication lab, and an addition lab. In the addition and multiplication labs, students must supply the lab technicians with the number which they wish to add or multiply. The technicians carry out the students' instructions. In the graph lab, students are presented with three graphs, only one of which is correct. After examining the graphs, the solution may be obvious.

While students may take as much time as they wish before making a move, it is to their advantage to take as few moves as possible and to consult the labs only when they really need help. Students are encouraged to do as much of the work as they can using paper and pencil.

The problems are graded in difficulty within a level, and those which are more challenging are presented when a student demonstrates that he or she is proficient at a given sublevel. A student who cannot solve the three equations presented at a sublevel before the bomb explodes is sent to an easier sublevel.

Levels are designed to reflect the course syllabus and include the solution of linear equations, quadratics, systems of equation, and trigonometric problems.

Student Programing in LOGO

How many times have students asked you to show them a solution "step-by-step" or confided that they "just can't think logically?" For students who until now have lacked confidence in their ability to analyze mathematical problems, we believe the process of writing programs for simple, commonly-used algorithms--such as solving linear equations, applying the quadratic formula, finding the discriminant of a matrix--should help to restore confidence. The process of writing programs should also reinforce concepts behind the algorithms, and should enhance a student's ability to explain how the answer was

obtained--to deliver the procedure as well as the end result when solving problems.

The challenge is to introduce programing in such a way as to minimize time spent on programing skills and maximize the mathematical benefit. This point is especially crucial in an algebra-trigonometry course--which essentially represents a student's last chance to acquire facility in a wide range of areas before calculus. (At an engineering school such as ours, this knowledge will typically be applied in differential equations and linear algebra as well as the full calculus sequence.) In light of the nature of this course, it seems appropriate to introduce programing slowly, integrating instruction in small amounts within the larger mathematical context rather than having a "programing unit." For this approach to be effective, the programing language used must be easy to learn as well as mathematically versatile.

We were sorely tempted to choose BASIC as the language for the course. Developed at Dartmouth in 1965, BASIC is relatively easy to learn, versatile enough for the needs of the course and, most importantly, because of its availability on virtually all micros as well as most other computers, BASIC has become a kind of universal language. However, in our opinion, students at the entering level of this course would find using BASIC difficult, particularly the looping and branching procedures. It is noteworthy to observe that typical college-level courses involving any significant degree of BASIC programing have algebra as a prerequisite.

The LOGO language, originally developed at MIT in 1968 using mainframe computers, is now widely available on micros thanks to Seymour Papert, Harold Abelson, and others. (For an excellent introduction to LOGO, see Abelson 1982.) LOGO shares many of the positive attributes of BASIC. Like BASIC, it is interactive (a highly motivating factor) and it is mathematically versatile. And "turtle geometry," for which LOGO is famous, facilitates graphics to such a high degree that it is even used successfully with very young children. (See Abelson & diSessa 1981, for applications to higher mathematics.)

However, the primary advantage of LOGO for our purposes, in terms of ease in learning and in mathematical application, is the procedural structure of the language. In LOGO, a program can call any procedure already defined simply by specifying its name; in this way the student can build up a library of primitives, any of which can be referenced in any new procedure. Furthermore, a procedure can reference itself (recursion), which enhances the creation of loops. In particular, simple loops can be constructed using "tail-end" recursion--referencing the procedure itself as the last step of the procedure. Conventional loops are also possible, using the REPEAT statement.

Generally speaking, just as the typical BASIC program consists of a series of executions of subroutines, each referred to by a statement number, the typical LOGO program consists of a series of procedures, each referred to by a procedure name. The clarity thus achieved is substantial; not only can students break down the algorithm in question into steps, but they can refer to each step by its name.

In conclusion, the innovative and varied approaches to using microcomputers for this algebra-trig course were implemented beginning in September 1983. This pilot course is being subjected to rigorous evaluative procedures, and a five-hour, noncomputer section of the course has been run simultaneously as a control. Plans are being considered to include in future experimental sections a component of computer management as well.

REFERENCES

Abelson, H., LOGO for the Apple II. New York: BYTE/McGraw-Hill, 1982.

Abelson, H. and A. diSessa. Turtle Geometry: The Computer as a Medium for Exploring Mathematics. Cambridge: MIT Press, 1981.

Ihor Charischak

RELEARNING MATHEMATICS THROUGH LOGO

Like it or not, mathematics remains a great "unloved" subject in school. In fact, its dislike is so prevalent it has become socially acceptable to admit one's fear of or inability to do mathematics. One reason for this aversion is that few people really sense any satisfaction from the study of math. According to Seymour Papert, author of Mindstorms (1980), this malady is a result of the student's failure to experience what is at the heart of the mathematician's work: the process of discovering mathematical truths through open-ended, exploratory activities. Unfortunately, curing the malady is not an easy task. What promises to make an impact on this problem is the creative application of a variety of software tools. This paper describes how one of these tools, LOGO, can improve the quality of mathematics education.

LOGO: An Electronic Learning Environment

LOGO was developed at MIT in 1968 under the leadership of mathematician Seymour Papert for the purpose of giving people a vehicle for exploring mathematics and developing their thinking skills. LOGO can be characterized in four ways:

First, the student is in control, not the teacher. The student controls an electronic robot-like "turtle" (which appears as a small triangle in the middle of the computer screen) with the commands: FORWARD, BACK, RIGHT, and LEFT. As the turtle moves, it draws a line with its "pen." By rotating the turtle and drawing more lines, interesting visual displays can be made. Secondly, learning evolves from the student's own needs. As the user explores the screen environment, unexpected things usually happen. For example, the turtle may disappear off the screen. The student would then ask the teacher or another student for help. By solving this mystery, the student gets insights which suggest further exploration. In this way, the student learns powerful new ideas about the LOGO environment. Thirdly, mathematics is learned informally. Since LOGO is a very mathematically rich environment, the user masters this world by exploring mathematical ideas without necessarily knowing the labels. For example, the student drawing a square learns that a RIGHT or LEFT 90 is needed. Now the two labels we can attach to this command are: (1) The unit for measuring angles is the degree, and (2) 90 degrees is a right angle. This is certainly useful information but it does not have to be introduced at this point. A more appropriate time might be during math class when students can draw on their experience with

---

Ihor Charischak, Computer Education Coordinator, Board of Cooperative Educational Services (BOCES), Westchester County, New York, is a former mathematics teacher and computer coordinator for grades 1-12.

---

the turtle to understand these concepts. Finally, LOGO is a more natural way to learn some powerful mathematics. For example, the way angles are introduced in the typical classroom is to have students draw a pair of line segments (or rays) and measure the gap between them with a protractor. Usually the student is asked to accept on faith that this is a useful activity, but in the LOGO environment, one needs to know about angles to make the turtle do something useful.

What follows are examples of four LOGO lessons where the goal is to have students discover mathematical truths.

LOGO Discovery Activities

Lesson 1: The Total Turtle Trip Theorem

This theorem states that there is a relationship between the number of turns and the size of the turns in drawing regular polygons. (A regular polygon is a closed shape where all sides are equal and all interior angles are equal.) The activity asks each student to draw a 3-, 4-, 5-, 6-, 8-, and 12-sided regular polygon using the sentence:

```
REPEAT n  [FD 20 RT a]   where n is the number of turns
                         and a is the size of the turn
```

A summary of the exploration may look like this:

Command	Resulting Turtle Drawing
REPEAT 3 [FD 50 RT 120]	Equilateral triangle
REPEAT 4 [FD 50 RT 90]	Square
REPEAT 5 [FD 50 RT 72]	Regular Pentagon
REPEAT 6 [FD 50 RT _?_]	Regular Hexagon

The main question to answer here is: What is the relationship between the repeat number and the turning number for each polygon? (Answer: The product of the repeat number and the turning number is 360.) This makes sense if you realize that the turtle has to make a complete turn in order to draw a polygon. For example, a hexagon is drawn by having the turtle repeat 6 times: move forward a certain amount, and then turn 60 degrees.

Lesson 2: Discovering Circles

Notice that when the polygon increases its number of sides, the polygon appears to be more circular. In fact, the command statement that produces a turtle "circle" is: REPEAT 360 [FD 1 RT 1] which is really a 360-gon. Contrast that definition with the more commonly known Euclidean model: a circle is an infinite set of points all of which are equidistant from a given point. Also compare the turtle model to the Cartesian model: $X^2 + Y^2 = R^2$ where X and Y represent points along a pair of number lines, and R is the radius. The turtle model is simpler to comprehend and can be used to introduce the student to the more powerful ideas of the calculus.

Lesson 3: Discovering π (Pi)

An interesting relationship in geometry is that the circumference of a circle when divided by its diameter is always the same no matter what size circle is being measured. This ratio which is called π turns out to be constant and approximately equals 3.14159. Probably what confuses most students is understanding that a mixed number can be a ratio. All that is being said is that the circumference of a circle is always (3.14159...) times bigger than the diameter. One activity that is effective in getting students to understand that C/D is the same value for all circles, asks the student to measure circular objects. This activity is difficult to do in the real world, but not in a LOGO environment. When the turtle draws a circle, it is easy to measure. For example, REPEAT 360 [FD 1 RT 1] produces a circle with a circumference of 360. To find the diameter, we turn the turtle 90 degrees and estimate its trip through the diameter. If our estimate is too big, we can erase and back up until we get the closest approximation for the diameter. So if our circumference is 360, then our diameter will be approximately 115. By repeating this experiment with different size circles, the student should soon see that the circumference is always a little more than 3 times bigger than the diameter. Later when it is appropriate, π can be introduced.

Lesson 4: Discovering The Area of Circles

In the turtle environment, the area of a circle can be investigated without using π. It turns out that the formula A = 1/4 x C x D where C is the circumference, D is the diameter, and A is the area, is a more convenient formula than the better known version: $A = \pi R^2$ where A is the area and R is the radius. This formula is an extension of a little known theorem for finding areas of polygons: A = 1/2 x Apothem x Perimeter.

Conclusion

As these activities suggest, the use of LOGO in mathematics instruction can affect the quality of mathematics education. It is a powerful way to expose students to the satisfactions of mathematical exploration, which is the foundation for success in the study of mathematics.

REFERENCE

Papert, Seymour. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc., 1980.

Let me consider a second kind of technique--a technique associated with position. In mathematics it is essential that the position of components within a mathematical expression be observed. Components must be in their correct places. Indeed, we formulate fundamental mathematical principles such as the commutative laws to allow us, under certain conditions, to alter the position of information--of bits within bytes--without altering the information conveyed.

This is not true in ordinary speech. It is not necessary that a particular element of speech always appear in a certain position within a sentence. There are alternative positions that do not result in altered meaning.

In computer communication, the location of an information item is absolutely critical. As every programmer knows, matters of syntax must be carefully observed. As a basic skill, students will need to observe the importance of sequence and position in communication. One essential difference between computer communication and vernacular communication is the paramount role of position in a computer environment. Somehow we will need to emphasize this difference in the teaching of reading, mathematics, and computing in the educational programs designed to prepare students for life in the computer culture of the next century.

One of the things that certainly must receive educational attention is what the computer scientist refers to as "terminators" or "separators"--punctuation used to indicate the termination of a communique. Those of you who are experienced programmers know the importance of commas, spaces, and semicolons. The role of terminators and separators in the computer environment will need to be reflected in the educational activities that prepare students for that environment.

There are two more aspects or technique of operation that I wish to consider. One may be familiar to you; the other, I think, will be unfamiliar and perhaps intriguing.

The first is the need for increased awareness of identifiers. In a computer environment, data occur in several forms. The ways in which data forms are identified are similar but not identical to identification techniques used outside the world of computers. Explicit identifiers are used in computer technology much as they are used in vernacular speech. We all recognize the date "January 1, 1984" because it is explicitly identified by the format of the communication. In a computer environment, explicit identification is somewhat more difficult because the coding--the way in which the information is transmitted--may not be sufficiently varied to convey to the operator that the code represents a date. We will need to provide more experience in identifying coded statements explicitly, because the ability to recognize a data form is a fundamental skill within a computer environment.

In vernacular speech many data forms are identified implicitly by their context. In one context a number within a sentence may represent a year or date. In another context it may represent a quantity--a number of people. In yet another context the same number may represent an amount of money. The identifica-

tion may be implicit rather than explicit. We will need to develop the student's ability to identify data forms from contextual relationships since, in many computer environments, contextual identification is the only means of identification.

In fact, data often are identified indirectly. That is, the information that is provided the operator is not the information that is required but rather information that points to the "address" where the required information resides. This is a means of identification unique to a computer environment, but it will be increasingly used as our world and the computer world coincide. Educators will need to find ways of teaching the use of "pointer-data types"--information that points to other information, rather than providing the information directly--since the use of pointers certainly will be fundamental in a future computer environment.

The last technique that seems to me to be of interest is quite different from techniques that we now consider basic. It is the technique associated with the computer notion of a default. A default in a computer environment is the computer's capability to supplement possibly incomplete information provided by the operator, with information that already exists within the system. Educators will need to provide students with experience in utilizing the default capabilities of computers in systematic ways to enhance the efficiency of data transmission. The use of defaults and the psychology of defaults will be a fundamental skill in the computer environment of the future.

Let me move to the second category, that of concepts.

Students will need to understand in new ways how information is created, processed, and used, if they are to function effectively in a computer culture. What do students of today perceive as the origin of the information which they acquire in your classes?

Probably they view 90 percent of it as somehow transmitted by genetic coding. Most assume that the data base is static; that it existed in past times; that it is transmitted by the teacher to them; and, if the need arises, they will retransmit it to their children or students in much the same form as they received it. I think that this understanding of information is inappropriate in a computer environment. We as educators will need to retrain students to view information and data bases as dynamic rather than static. Information is something to which every user in a computer environment can contribute. Data bases can be accessed and supplemented. Data bases are constantly changing. A data base can be accurate or inaccurate because data that are dynamic are not invariably correct. The student's view--that if it's in the book, it's right--must be changed when the "book" becomes the computer. Not only will students need to have a new conception of the origins of information but also they will require new ways of understanding how information is processed. In the past, information was at best partially ordered. That is, if two items of information were compared, usually it was possible to determine that one item was prior to the other. But frequently it was not possible to arrange items linearly in any meaningful way.

In future information data bases, information may be linearly ordered. Data certainly will be discrete since those conditions are required if information is to be processed step-by-step. We will need to teach as a basic skill how information can be sequenced, and how it can be considered and processed, beginning with an initial item and proceeding through the items in the appropriate sequence. That kind of skill--for those of you who teach mathematics, and I suppose also for those of you who teach communication skills--is rare indeed in today's educational environment. The algorithmic, sequential, stepwise approach to the treatment of data is not a widely held competency in today's culture. It will be an essential skill in the computer environment of the future.

The uses of information also will change. Students will need to understand, as a basic skill, how the information that they acquire can be used to attack impending problems--to formulate personal decisions. In the past, acquired information was used by students and adults primarily to verify actions. Hunches led to the decisions of the past. Intuition, after all, is a primitive information-processing technique. While we may use information to verify whether the outcomes of our hunches are correct, we rarely use data explicitly in formulating hunches. In the computer environment of the future, the information will be useful in reaching decisions and not just in verifying them. The student will need to be trained, as a basic skill, to use information before the decision and not just after. For example, one important mode of decision-making will be simulation, that is, the replication of events without their actualization. Simulations create an artificial environment where the results of hunches can be verified *a priori*. Action based upon the hunch can be restrained until something is known of the potential outcome of that decision.

Let me move finally to the category of thought, of knowledge, of epistemology. It's my conviction that the very way in which we think will be changed by a computer environment. What we consider to be knowledge will change. The way in which we acquire knowledge will change. Somehow we will need to include educational activities that teach students new epistemological principles. I'm not sure that we now convey much of epistemology to basic-skills students, but I suppose implicitly we do. When we teach them to read, we are suggesting that knowledge is somehow incorporated in the written word. When we teach them to add fractions, somehow we are conveying that the way in which mathematical operations are performed is culturally important. Educators will need to provide the same kinds of experiences in a computer environment, but they will lead to different conclusions.

Three aspects of epistemology must be considered as we plan educational programs for the future. The first is definition. One way of knowing is to define--to subsume a number of ideas under one idea, using a definition. We teach definitions now in reading. We teach definitions in communications of all kinds. Certainly we attempt to teach definition in mathematics. Can the nature of definition be different in a computer culture? Are we so bound to our present epistemology because of some neurologic condition that we will never permit a new view of definition? I think not. The role and nature of definition in a computer environment will change substantially. The computer has the unique capability of giving a kind of "circular" or recursive definition. With the computer, a function can be defined to call itself; procedures can use themselves as subprocedures. This is unique to a computer environment.

Mathematicians have utilized recursion and recursive definitions in the past, but this will be an important mode of definition in the future for all computer-literate individuals. For example, the standard definition in many computer languages of N factorial, $N! = N \cdot (N-1)!$ clearly is a circular definition by former epistemological standards because it defines a concept in terms of itself. As another example, consider the following: A binary tree is a root with two binary subtrees. If a binary tree is not understood in the beginning, how can this definition be of aid? To a computer, a recursive definition is useful. In the computer environment of the future, recursive definition will be an important, if not a preeminent, mode of definition.

The second epistemological component, classification, will change as well. An important way of knowing is to classify. Mathematicians are great classifiers. Teachers of grammar are great classifiers. Much of knowledge is based upon our ability to classify. Some neuropsychologists claim that humans are limited in their classification ability because of our neurosystem. Piaget asserts that children at the preoperational and concrete operational stages have very limited classification abilities. Multiple classifications are not really practical before the onset of the formal operational stage. It is unusual even in an adult for triple classification to occur. On the other hand, the computer is an admirable classifier. Thus, in a computer environment, multiple classifications are possible even for the average individual. Educators will have to find ways to teach--as a basic skill--multiple classification. The ability to link data sets to other data sets through a variety of cross-classifications is absolutely fundamental. It may even be possible in the future to define creativity in terms of the extent and richness of the cross-classifications observed among data sets. If we can define creativity, can we not teach it? Can we not teach it as a basic skill in a computer environment?

Lastly, let me comment upon the epistemological skill of verification--of validation. Knowledge becomes truth only upon verification. Thus truth and verification are inseparably bound together. To understand truth, we must look to the mode of verification.

Let us consider three kinds of truth: truth for the person in the street, truth for the scientist, and, finally, truth for the mathematician.

The basis of popular truth is authority--what you hear on television; what you read in the National Enquirer or The New York Times; what the Bible says. Popular truth is based upon authority and belief.

What is scientific truth? On what basis does a scientist accept knowledge as truth--by objective verification, that is, the extent to which it matches observations. Empirical truth is obtained through scientific validation.

Finally, to the mathematician, truth is determined by consistency with assumptions. The mode of verification is proof--deductive proof. It is my belief that, in a computer culture, all three modes of truth--all three modes of verification--will be subject to revision, perhaps even rejection. Already we are seeing this change in the field of mathematics. The four-color map problem was solved not long ago using a case-by-case analysis via computer. While some

mathematicians have resisted that mode of proof, I believe that it will be tomorrow's predominant mode. The computer wave that only now is washing the sands of the foundations of mathematics certainly will rearrange them.

Scientific truth too will be changed by a computer environment. It will not in all cases be a comparison between what is thought to be knowledge and the real world, but rather a comparison of knowledge with a simulation of the real world. It will be truth relative to a computer model. That is already the case in many scientific areas. What is true depends upon the extent to which it matches the model. A model will be true, perhaps, to the extent that it matches reality, but there are portions of reality that we cannot know. For example, a computer model of creation could permit "verification" of knowledge both of our unknown past and unforseeable future.

But what about popular truth? Will human beings continue to rely upon truth determined by authority--by belief? I think we will, but the authority and the beliefs will change in a computer culture. Popular truth in the next century possibly will be determined by a computer description of our own reality. Belief in the omnipotence of the computer, and not in the omnipotence of a supreme being, a supreme politician, or a supreme newscaster, will be the basis of popular truth. We have some sense of this phenomenon already. If any of you have had the unpleasant experience of challenging a computerized bank statement, you will know precisely what I mean. As educators we must provide, as a basic skill, an understanding of the nature of computer truth. An understanding of truth is fundamental to any educational activity. Certainly the nature of truth in a computer culture will be fundamental to educational activities that prepare students for life in the next century.

My presentation has been but a glimpse of the computer culture of the 21st century, and the educational characteristics of that culture. I'm sure that many of you will not be convinced that this glimpse is accurate. You may say, "That's an interesting comment, but it's unlikely that those basic skills will replace really important things like adding fractions or spelling Mississippi." As educators, we must consider that what is important always is related to the tools which we use to attack a problem--to the way we approach a task. As we approach the tasks of the next century, using the computer technology that we are here today to examine in its infant form, we must reconsider literally all of our educational preconceptions if we are to be able to redefine basic skills for a computer culture.

John Bell

EARTH'S FULL COLORS

A day spent at CUNY's 1983 conference Microcomputers and Basic Skills, co-sponsored by Long Island University, left me feeling a strange mixture of exhilaration and dismay. Like a little boy--along with so many other little boys and little girls--I stood at the edge of Miranda's "brave new world," and like an adult--along with many other collegial adults--I felt Aldous Huxley's Brave New World rumbling under me.

I walk out of work each day, and the Brooklyn Bridge sits before me, a splendid reminder of America's, and the world's aspirations as mankind turned into the twentieth century. The Brooklyn Bridge conquered space, much as the car that soon followed it, and the plane, and the rocket ship. Pioneer I has just galleoned past Pluto, out where our sun looks like a distant and "real" star, and our Earth probe will wander, perhaps for billions of years, in relatively benign interstellar space. Mankind has smiled a good deal since the completion of the Brooklyn Bridge, but the twentieth century has also brought with it much tempest.

One speaker at the conference noted that anthropomorphizing computers makes them more effective. People who personify the boxes they use perform better with them. Another suggested that we may have to prepare for a culture in which belief in God is replaced by a belief in the omnipotence of a computer. These are by no means strange ideas. Nor are they new. But it is strange, for me at least, to hear them presented with such matter-of-fact acceptance, indeed almost youthful glee. And by men whose faces were not well-lit, and to audiences sitting in dimly illumined rooms. A voice and faceless faces. A bizarre world of benign Francis Bacon's making. In a building of white, white, white walls and rooms filled with flashing blank television screens, ready to open their windows into the colorful world of CAI.

There were green floors, it is true. And a few paintings and large dolls in the lobby, but mostly the tones were harsh; and stark blacks and whites dominated. I felt no presence of browns. No soil. No living things. At the end of the conference, a warm and smiling man sat alone playing the piano. No one gathered around, but a young woman passing by said, "Thank God for the music."

John Bell, Assistant Professor of Writing and Speech at New York City Technical College, has also served there as Director of The Writing Center.

GLOSSARY

Asterisk (*)	The multiplication sign in the BASIC language.
BASIC	An acronym for the computer language Beginner's All-Purpose Symbolic Instruction Code. Developed by Kemeny and Kurtz at Dartmouth in the 1960s. Generally considered easy to learn and commonly taught as a first language.
Boot	Short for "bootstrap." The process of loading any program into a computer's memory in order to operate the program.
CAI	An acronym for computer-assisted (or -aided) instruction. Any instructional application of computers, but particularly programs in the form of drill and practice, tutorials, demonstrations, simulations, or educational games.
CAT	Computerized adaptive testing.
CBT	An acronym for computer-based training, which incorporates both computer-assisted instruction and computer-managed instruction.
CEL	An acronym for computer-enhanced learning.
CMI	An acronym for computer-managed instruction. Primarily classroom management systems in which computers are used to help teachers organize and manage teaching and record keeping for classes.
Copy-protected	A disk which is difficult if not impossible to duplicate.
CPU	An acronym for central processing unit. The heart of the micro, which coordinates and controls the activities of all other units and performs all the arithmetic and logical processes.
Crash	To stop operating.
CRT	An acronym for cathode ray tube. A video screen or monitor used for display.
DIR	Directory. On a disk, a system file indicating the names and locations of all files contained on the disk.
Disk drive	A mechanical device for running programs stored on disks.
DOS	An acronym for disk operating system. A program which manipulates the information on a disk.

Dot-matrix printer	An output device which prints characters and graphics on paper using dot configurations.
Download	To transfer data from a central computer to a smaller computer, as from a mainframe to a micro.
Extendable	(Also extensible.) Said of a computer language which can be modified by the user to create new programing commands in terms of old ones.
HI-RES	An abbreviation for high-resolution, a kind of graphics display yielding sharp images.
Interactive	The characteristic of a system which allows the user to input data and to receive immediate feedback.
Joysticks	See paddles.
Light pen	A pen-like device connected by cable to some CRTs. It is used to draw, point to, move, or delete images on the screen.
LOGO	A computer language developed by Papert and others at MIT in the 1960s. Because of the ease with which it is learned and its striking graphic capabilities, it is commonly taught to children.
Menu	A list of options which a program presents to the user.
Mainframe	A large, powerful, multiuser computer generally accessed by means of terminals. Contrasts with a minicomputer (the size of a refrigerator) and a microcomputer (the size of a typewriter).
Master Diskette	A disk that can be booted simply by inserting it into the disk drive when the computer is turned on.
Paddles	Mechanical devices connected to the computer which can be used to move objects around the screen or to manipulate data. Generally associated with arcade games.
Pascal	A computer language named after the French mathematician Blaise Pascal and developed by Wirth in the early 70s. Designed so that programs written in that language can be debugged easily.
PLATO	An acronym for Programed Logic for Automated Teaching Operations. A nationwide software delivery system involving mainframes. Developed in 1959 at the University of Illinois, it offers access to a large number of college-level programs. Some PLATO software is now available for use on micros.

Primitive — A computer language command in terms of which others are defined. (See extendable.)

Procedural — Said of a computer language which possesses the capability of expressing commands in a form similar to "natural" languages such as English or mathematics.

RAM — An acronym for random access memory. The term refers to memory in the computer which can be altered or erased. The more RAM a micro has, the better.

Recursion — The concept of defining a process in such a way that each repetition depends on the immediately preceding result. An example of recursion: Suppose that you decide that the size of your child's daily allowance during a given month will be 1¢ for the first day of the month, and for all other days of the month, double the previous day's allowance.

Slave Diskette — A disk that can be used only after a master diskette has been booted.

Unlistable — Said of a program the steps of which the user cannot get the computer to list

LIST OF CONTRIBUTORS

Baruch College, CUNY
17 Lexington Av., NYC 10010
GERARD M. DALGISH (English)

BOCES Southern Westchester
Concord Rd., Ardsley, NY 10502
IHOR CHARISCHAK (Instructional Services)

Borough of Manhattan Community College, CUNY
199 Chambers St., NYC 10007
GEOFFREY AKST (Mathematics)
MARVIN KUSHNER (Data Processing)
REINHARD KUSSAT (Testing)
LAWRENCE SHER (Mathematics)

Bronx Community College, CUNY
University Av. & W. 181 St.
Bronx, NY 10453
ELLEN K. LEAHY (Learning Center)
ANNETTE PERETZ (Learning Center)

Brooklyn College, CUNY
Bedford Av. & Avenue H
Brooklyn, NY 11210
D. TERENCE LANGENDOEN (English)

Christopher Newport College
50 Shoe Lane, Newport News, VA 23606
ALICE F. RANDALL (English)

The City College, CUNY
Convent Av. & 138 St., NYC 10031
JOHN C. MILLER (Mathematics)

The City University of New York
535 E. 80 St., NYC 10021
MAX KIRSCH (Academic Computing)
MARIE J. LEDERMAN (Acad. Affairs)
RITA TERDIMAN (Academic Computing)

Cornell University
375 Olin Hall, Ithaca, NY 14853
ANNE B. GRINOLS (Reading & Study Skills)

Hostos Community College, CUNY
475 Grand Concourse, Bronx, NY 10451
CLARA VELAZQUEZ (ESL, English)

Jersey City State College
2039 Kennedy Blvd.
Jersey City, NJ 07305
ROBERT LATZER (English)

LaGuardia Community College, CUNY
31-10 Thomson Av.
Long Island City, NY 11101
MARIAN ARKIN (English)
BRIAN GALLAGHER (English)
JOAN GREENBAUM (Economics)
DANIEL LYNCH (English)
KAREN PEARL (Counseling)
ESTELLE SCHNEIDER (Communication Skills)

The Johns Hopkins University
Baltimore, MD 21218
BERT F. GREEN, JR. (Psychology)

New York City Technical College, CUNY
300 Jay St., Brooklyn, NY 11201
HENRY AFRICK (Mathematics)
JOHN BELL (Writing & Speech)

New York Institute of Technology
1855 Broadway, NYC 10023
also
Old Westbury Campus, P.O. Box 170
Old Westbury, NY 11568
JO ANN COMITO (Instructional Development, Old Westbury)
MERYLE KOHN (Mathematics, NYC)
ANNA SILVERSTEIN (Mathematics, Old Westbury)

Oakland University
Rochester, MI 48063
HELEN J. SCHWARTZ (English)

Pennsylvania State University
Fayette Campus
P.O. Box 519, Rte. 119 North
Uniontown, PA 15401
ROBERT C. GILLINGHAM (English)

Pittsburg State University
Pittsburg, KS 66762
ELWYN H. DAVIS (Mathematics)

Queensborough Comm. College, CUNY
222-05 56th Av., Bayside, NY 11364
ELIE LOWIE (Physics)

Southeastern Louisiana University
University Station
Hammond, LA 70402
CAROLE McALLISTER (English)

Teachers College, Columbia U.
525 W. 120 St., NYC 10027
BRUCE VOGELI (Mathematics & Science Education)

Towson State University
Baltimore, MD 21204
LINDA J. ALLRED (Psychology)

University of California, Irvine
Irvine, CA 92717
ALFRED BORK (Information & Computer Science)

University of Miami
P.O. Box 248127
Coral Gables, FL 33124
WILLIAM R. OATES (Communication)

York College, CUNY
150-14 Jamaica Av., Jamaica, NY 11432
JOAN BAUM (English)

BEST COPY AVAILABLE